Performance improvement of the proxy servers' cache in distributed systems using session and cookie techniques based on game theory

Bita JAFARI, Peyman BAYAT*
Department of Computer Engineering, Rasht Branch, Islamic Azad University, Rasht, Iran
jafari_bita@yahoo.com, bayat@iaurasht.ac.ir

***Corresponding author:** Peyman BAYAT
bayat@iaurasht.ac.ir

Abstract: Smart machines achieve daily life in-demand services increasingly, and quicker access to required services is of great importance and, in some services, is very critical. In this paper, a new architecture is proposed to improve the distributed cache proxy server performance in distributed systems for demanded services by developing cookie-session middleware and employing polymorphism mechanism in DHT servers. By using node clustering, each node as a peer is a server/client that listens to the line on the TCP port and uses session-cookie middleware for the requested service. When it is only a node, it is a server/client that listens to the line on the UDP port and performs DHT protocol for registration of the peers' characteristics. An architecture based on the game theory has been proposed in this research. Moreover, an extensive-form game with imperfect information, called weak sequential equilibrium, has been studied and proven. It has weak sequential and Nash equilibriums in the state of ((m_{tcp}, m_{udp}), (Y, Y); λ =1, μ=0)). The requests prioritization in the cookies has been studied, and load balancing of the sessions has been performed based on Round Robin prioritization. The simulation results show that the new architecture is optimal in terms of bandwidth consumption, user response time, and cache memory hit ratio.

Keywords: proxy server, cache distributed, session and cookie, Game theory, Weak sequential equilibrium.

1. Introduction

In research and industry communities offer different solutions for storing web in file-sharing systems and quick access to the resources and their allocation. Some of these methods are deploying memory servers at the edge of ISP, cache increment in routers, switches, servers, and creating a hierarchical cache system (Li et al., 2017; Tanenbaum, 2012). These are costly methods and need high investment to purchase and install the equipment and employ specialized personnel and result in a single point of failure. Since all peer-to-peer traffic passes through a proxy, when entering lots of requests simultaneously, responses to the requests are slow. On the other hand, the memory of the proxy is unlimited, and the size of the shared audio and video files is more than hundreds of megabytes. Such a storage structure is the main disadvantage of peer-to-peer networks that makes the scalability and accessibility of the networks controversial (Manesh et al., 2014; Song et al., 2017). The distributed cache proxy server provides high volume data transfer among some users directly, without using a central server, and with the least bandwidth and speed. By this protocol, websites can provide their users' required information from other users. Indeed, this protocol minimizes the need for a central server and provides data transfer among users directly. In this method, by developing the cookie-session middleware and polymorphism mechanism in the distributed cache memory, minimum bandwidth is consumed, and quick service access is achieved. The proxy server authenticates the users and connects them. Data storage and other properties of the distributed cache memory in cookie and session is achieved using distributed cache memory management middleware.

A session is a temporary storage that can be quickly accessed and is allocated to any machine intending to use the service and is destroyed by exiting and service termination. This temporary storage is located in the file system, database, or internal storage of the program executing the service as a text file. Each machine has a session ID for storing its data on a session, and can only access its relevant sessions. The session is stored both on the server and the client. If it is stored on the client, it is stored as a cookie, and if it is on the server, session IDs are created and managed by the server. Hence, if millions of users have access to the server, unique session IDs are generated and managed by the server for each user (Li, 2017; Mun & Hyesook, 2017; Sumalatha et al., 2016; Tanenbaum, 2017).

https://doi.org/10.33436/v31i3y202104

The new architecture is not only innovated for transferring multimedia files, but also for providing the demanded services, such that cache memory is managed using the session-cookie technique, and polymorphism mechanism is used in the distributed cache memory as one of the crucial aspects of the distributed object-oriented system, and examined using Nash equilibrium game theories in such a system, that reduced the bandwidth consumption and service response time, and increased the cache memory hit rate.

2. Related works

A cache is used for quick access to the resources. It shortens the route between web servers and the customers as an intermediate server by storing the resources. Moreover, it results in the network service improvement by reduction of the response time, network traffic, and bandwidth (Jiang et al., 2014). Access methods for a distributed cache as the methods of service detection include centralized methods, based on supernode, DHT, flooding, caching, and RIC (He, 2017) (He et al., 2017). Unstructured peer-to-peer service detection based on flooding is the other method that needs memory for saving routing information. But, it makes heavy traffic in the network, especially when several queries of service detection are performed that make the network congested and results in delay (Moeini et al., 2017). In the store based method, information is not addressed, and most of the useful information is stored in the cache. So, it may use lots of the cache volume for more efficiency. Thus, it is not suitable for devices with limited space and doesn't consider mobility (Li et al., 2017). In the RIC-based method, static and mobile IoT devices with low memory and processor power are considered. In this method, each peer saves routing information in the cache for handling the routing service detection query. RIC space is limited according to the space of the IoT device. So, HBFI-based indexing is used to create an efficient space and demonstrate the capabilities of the IoT device, that is flexible in a combination of the abilities and removing the existing abilities (Gonzalez et al., 2017).

In the architecture of clustered physical cache, there are centralized and non-centralized strategies. In the centralized architecture, a unit controller receives all incoming requests then selects the cache server for the requests responses. This architecture is not scalable in large environments. In non-centralized architecture, there are several controllers in the system that incoming requests transfer to the controller or the next controller randomly. The non-centralized strategy is more scalable than the centralized one, but it needs more communication costs for sharing information between the controllers (Dudykevych & Nechypor, 2016; Li & Li, 2014).

Considering the popularity of social networks, web hosting, and video stream site, lots of researches are performed on the storing architecture of web contents so that proxy caches can perform requests of thousands of users for optimization of the contents. World wide web suffers from a lack of scalability and reliability results of overflow and accumulation of the servers [9]. The hierarchical and distributed architectures are integrated for better performance and efficiency that construct the hybrid architecture. In this architecture, a distributed cache is stored among several cache servers. load-balancing is the goal of the distribution mechanism of the stored versions in the distributed servers. Autonomous decision making based on cache management is a suitable solution for dynamic and self-compatible load balancing (Johnston, 2015). A cache proxy management framework based on autonomous elements for distributed cache systems of proxy is proposed in [6] with studying autonomous management technology and proxy cache management according to an autonomous decision. This architecture is a combination of modules of data monitoring, independent understanding, automatic decision making, and automatic adjustment. In this mechanism, the URL address is distributed uniformly among the cache nodes through the storing mechanism of the hash loop with virtual nodes. Cache status adjustment strategy is performed according to the historical sequence of a gray prediction model. For load reduction of cache nodes, a hot ranking of contents based on the stored domain name is predicted. The migration algorithm is proposed for dynamic migration of hot points of cache and merging virtual nodes on the hash loop (He et al., 2017; Sumalatha, Priyanka & Ahana, 2016).

A distributed system is a new model for presentation, consumption, and delivery of information technology services include hardware, software, information, and other shared

resources of cloud computing utilizing the internet or intranet. In such a system, users access the resources based on their demand and without considering the service and method of data delivery. A distributed system is a set of independent computers that users seam it as a coherent and single system (Dias et al., 2018; Haghi, 2017).

In (Zhang et al., 2020), in the smart transportation system, the data usually represents the users' session and demands. The traditional approaches are focused on data sequence, and the last item clicked by the user, and does not illustrate user priority correctly. A session-based aware multi-purpose model is suggested for the transportation system, which considers user behavior comprehensively from different aspects and utilizes an efficient and concise self-attention converter style to analyze the current session data sequence for a precise understanding of the user purpose. In this regard, the MASR model is comprehensive and efficient and is analyzed by considering some aspects such as general interests of the users, selective features, current user interests, user's long-term priority, and other items. Consequently, the recommended system responds to the user faster with more accurate results and provides more secure and smart transportation services.

In (Mistry et al., 2015; Song, 2015), a load balance strategy is suggested for improving the cache performance in homogeneous and heterogeneous proxy cache cluster (PCC), which adjusts the performance rate by combining various performance parameters. The independent data and virtual nodes (VN) connected to the cache node are generated via the random function using the oscillating weight data method.

In (Wang et al., 2020) by developing the communication networks quickly, the data interaction between heterogeneous networks such as the internet of things (IoT) and vehicle ad-hoc networks (VANET) becomes more common day by day. In adjacent devices cellular networks, files can be shared directly without using eNBs, which are called device-to-device (D2D) communication. This type of communication is a potential technological component in the future generation of communications. The traditional concentrated network architecture cannot run such user requests due to heavy load in blackball links and long delay.

Based on (Gibbons, 1992; Jung, 2015), game theory is a subset of mathematics that tries to predict the action and the decisions of the creatures with a right to choose, in interaction with each other using design and analysis of the scenario. Using game theory in the modeling based on the extensive game with imperfect information (weak sequential equilibrium), a player is not exactly aware of the game payoffs, or it doesn't know its competitor type. Equilibrium concept in the extensive game is weak sequential equilibrium, and its result is an assessment. In Nash equilibrium, a profile strategy is defined, but in weak sequential equilibrium, a belief system is needed in addition to profile strategy. It means that the decision-maker belief is formed where it doesn't know. So, a weak sequential equilibrium has two characteristics. The first one is sequential rationality that the decision-maker action, at each point, is optimal, based on all the events the players' payoffs are maximized, and there is no motivation for violation. The second one is of weak consistency. Meaning that the decision-maker beliefs formation is based on Bayes' rule that is the logical method of expected probabilities formation (Gibbons, 1992; Osborne, 2003; Osborne, 1994).

In this paper, the performance improvement of the proxy servers' cache in distributed systems is modeled in the extensive game with imperfect information using session and cookie techniques. Moreover, the weak sequential equilibrium of the players is investigated. Signaling game is one of the extensive games with imperfect information. In the nature signaling game, two different types are determined by two probabilities. The probability is a common knowledge that player 1 knows the probability, but it doesn't know the precise nature game. Player 1 is aware, but player 2 is unaware. Player 1 sends the signal (m1, m2) in a determined type to player 2.

3. Model, mechanism, and architecture of the proposed distributed cache

In this research, the distributed cache is a site that communicates with the proxy server as a middleware, interface, and site that polymorphism mechanism is identified for the creation of DHTs and the classes of session and cookie generation. The proxy server communicates with a

database including, the gathered information on the cookies, sessions, and the requests. According to the type of request the cookie is created, the session announces the readiness, the best session is selected for service, and a peer-to-peer connection is established between the client and the server.

3.1. Mechanism of the proposed distributed cache

The customers' clustering is performed based on the geographic region and the type of service. Using the proxy server and On-Demand feature, the presentation of the services by the cluster nodes is important. The request location and how to respond is not important. In figure 3.3, each of the cluster nodes 1, 2, 3,…, M presents service1, service2, service3,…, and service M respectively and they are as servers. Sometimes, these nodes need to service, and they are waiting for service as a client.

Load balancing is performed on the sessions of the proxy. The cluster nodes send their request to the proxy server, and they are service-oriented. The proxy server that is in the status of On-Demand, and always listen to the line, creates a cookie based on the class of cookie generation and the session. It establishes the connection between the node transmitting the request and the node presenting the special service in a peer-to-peer manner.

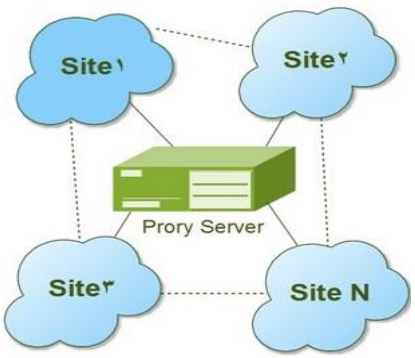


Figure 3.1. The relationship of distributed cache as a site with the proxy server

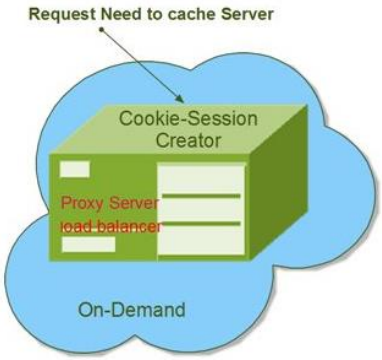


Figure 3.2. On-demand request to the distributed cache and proxy server

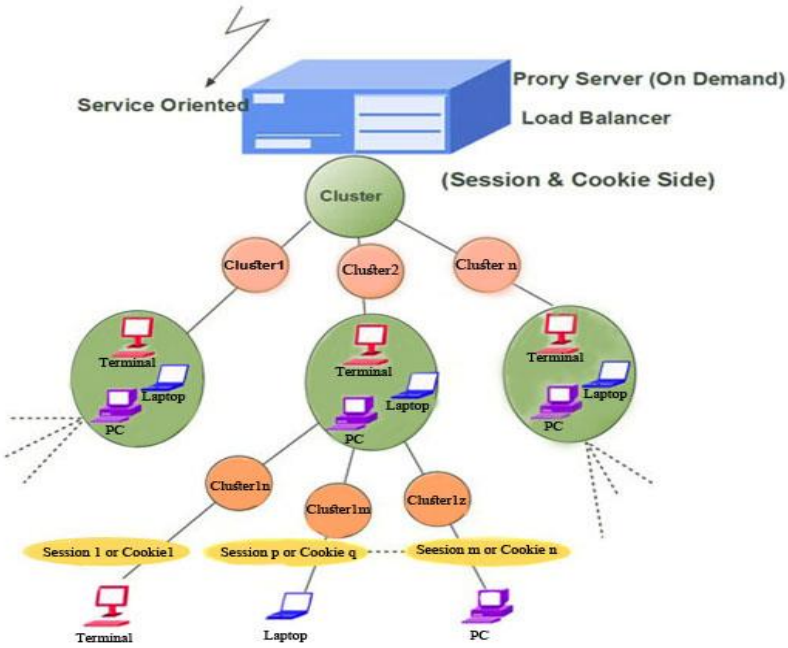


Figure 3.3. Communication of clusters and proxy server

3.2. Extensive-form game with imperfect information on the cache of the proxy servers in distributed systems

The review model investigates the efficiency of the proxy servers' cache in distributed systems using session and cookie techniques based on the extensive imperfect game. In this architecture, service discovery of distributed cache is performed by a combination of the clustering methods based on geographic divisions and DHT server. A node is both a client and a server. Prioritization of the request is investigated in cookies, and the preferred request is serviced, sooner. Load balancing in the sessions is performed based on the Round Robin method. A set of different types includes: $\Theta=\{ S_H , S_N\}$

In a cache, S shows the current state and service capability of the node. The larger value of S means that the cache node is busy, and the available service capability is low. Also, a smaller value of S means the free cache node and its availability with higher service capability. According to the following equation, the average of the current state of all of the nodes is calculated with notation S. $S = \frac{1}{n}\sum_{i=1}^{n} S_{current\,01}$

Two states of the cache node are as follows:

For cache node of i, if (i=1,2,3,…,n) $S_{curreent01} \geq S$, then i is a hot spot.

For cache node of i, if (i=1,2,3,…,n) $S_{curreent01} < S$, then i is normal spot.

The set of hot cache node is defined by H. $|H| = n_1$

The set of normal cache node is defined by N. $|N| = n_2$

The result of their sum is as the following: $|H| + |N| = n_1 + n_2 = n$

All the elements of H are sorted descending. S_H is the degree of the hot cache nodes business. The first node is the busiest one with lower service capability.

$S_H=\{ S_{H01}, S_{H02},\ldots S_{H0n},..\}$ $(S_{H01} \geq S_{H0(i+1)}$ $i=1,2,\ldots,n_1)$

All the elements of N are sorted ascending in terms of the state value. In the resulted sequence, the first node is the least busy one with higher service capability.

$S_N=\{S_{N01}, S_{N02},\ldots,S_{H0n},\ldots\}$ $(S_{H01} \leq S_{H0(i+1)}$ $i=1,2,\ldots,n_1)$

The proxy server has the role of nature. A node of S_H or S_N is selected for the received request, based on polymorphism and the middleware of the session and cookie. The selection probability from S_H or S_N sets constructs the probability set with a discrete probability distribution function and the sum of one. $P=\{p_1,p_2\}$

Probable P_k of the transmitter with type Θ_k is as the following:

$0 \leq p_k < 1$ and $\sum P_k = 1$

In this event, a set of the transmitter messages is transmitted from the aware player and received by the unaware player. Each node as a peer is a server/client that listens to the line on TCP port and utilizes Session-Cookie middleware for the requested services. But, as a node, it is a server/client that listens to the line on UDP port, performs DHT protocol, and registers the characteristics of the peers. So, player 1 transmits the message on TCP or UDP port.

$M=\{m_{tcp}, m_{udp}\}$, The set of actions of player 2 is that it is a service requester or receives the required service using the Session-Cookie middleware. Player 2 provides its characteristics using the DHT protocol for player 1 to use for future services. $A=\{a_1, a_2,\ldots,a_l\}$

The utility function of the transmitter or receiver player is the function of Θ_k, m_h, and a_l as the following: Payoff Sender: $u(\Theta_k, m_h, a_l)$ or $u(\Theta, m, a)$, Payoff Receiver: $u(\Theta, m, a)$.

The probability that the normal node replies to the service request is μ, and the opposite point is $1-\mu$. Moreover, the probability that a normal node performs DHT protocol is λ, and the opposite point is $1-\lambda$.

$Prob(S_N \mid tcp) = \mu$, $Prob(S_H \mid tcp) = 1-\mu$, $Prob(S_N \mid udp) = \lambda$, $Prob(S_H \mid udp) = 1-\lambda$

Profile strategy is defined as follows: $\{(tcp_{SN}, udp_{SH}), (a_{session\text{-}cookie}, a_{dht}), \mu, \lambda\}$

(tcp_{Sn}, udp_{SH}) and $(a_{session\text{-}cookie}, a_{dht})$ are the actions of player 1 and player 2, respectively, and λ and μ are the parameters of the belief system. It is an assessment that creates a weak sequential equilibrium. There are four states for the action of player 1. (m_{tcp}, m_{tcp}), (m_{udp}, m_{udp}), (m_{tcp}, m_{udp}), (m_{udp}, m_{tcp}). The strategy of rows 1 and 2 that signal is state-free, is called pooling strategy, and the strategy of rows 3 and 4 that the signal matches the game state, is called separating strategy. Equilibrium point should be found in the points that the player wants to make a decision, and it doesn't have any motivation for violation. To this aim, a pour strategy is defined from Θ space to m space, shown with б notation. б: Θ → m.

There is a pure strategy for the receiver from the received signals to the performed actions.

ρ : m → A, Generally, б and ρ form an equilibrium. First, б is optimized by giving ρ, and the following equation is true for each Θ:U(Θ , б(A), ρ (б(Θ))) ≥ U (Θ, m, ρ (m)):∀m ε M.

For each type of Θ, the obtained utility from the pure strategy game is larger than the utility from another signal. Since the transmitter decides about the message, none of the Θ type transmitters is motivated for message transmission. Optimization in the receiver with V payoff and the conditions of Θ, m, and *a* is as the following: Payoff receiver is V(Θ, m, a).

The player knows the payoff by knowing m and a, and it decides without knowing Θ. So, the calculated probabilities should be expected. The belief system is not required for calculation of expected probability, so that:

$$(\mu(\Theta \mid m) = (\mu(\Theta_1 \mid m), \mu(\Theta_2 \mid m), \ldots, (\mu(\Theta_k \mid m)), \mu(\Theta_k \mid m) \geq 0 \text{ and } \sum_k \mu(\theta k \mid m) = 1$$

It is a vector with the probability that when the signal is m player type is Θ_1 and Θ_2. the probabilities should be positive or zero, and their sum is one. With these probabilities, it is tried to maximize the expected utility. Max $\sum_k \mu(\theta k|m)V(\theta, m, a)$.

It is investigated how the beliefs are gained? A belief system of μ is a set of beliefs that results from different points. The probability of receiving signal m1 with the opposite player type using Bayes' rule is as the following:

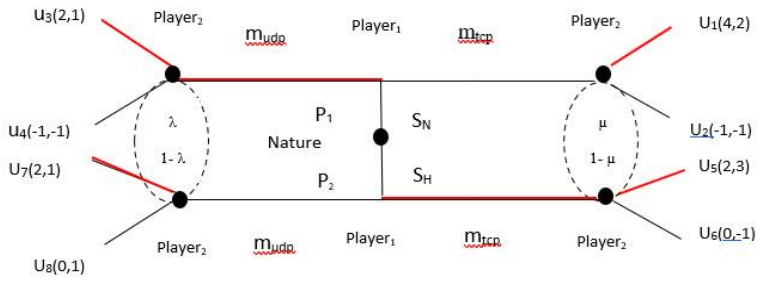
$$Prob(S \mid m_1) = \frac{prob\ (m1|s)prob(s)}{prob\ (m1|s)prob(s) + prob\ (m1|w)prob(w)}$$

In general, the followings equation can be extracted:

$$D(\Theta_k, m; б) = \begin{cases} 1 & \text{if } б(\Theta_k) = m \\ 0 & \text{if } б(\Theta_k) \neq m \end{cases} \qquad \mu(\theta_k \mid m) = \frac{\rho(\theta_k).D(\theta_k, m; б)}{\sum_k \rho(\theta_k).D(\theta_k, m; б)}$$

So, beliefs are constructed based on Bayes' rule. A player's belief matches another player's action. Hence, the player should act rationally in the actions and beliefs. Since they maximize the payoff, a belief should be coordinated with the action of the opposite player.

A) for state (m_{udp}, m_{tcp})



Graph 3.2.1. State (m_{udp}, m_{tcp})

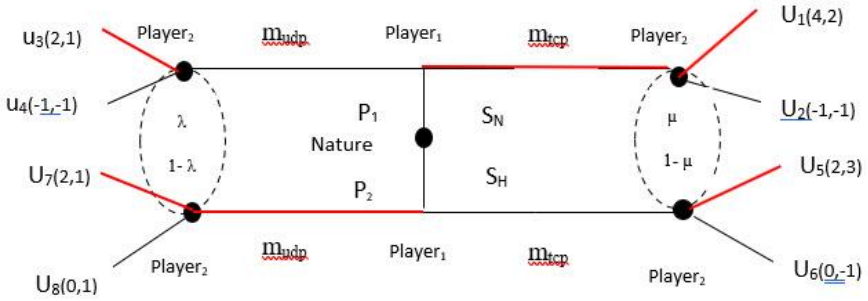
$$\text{Prob}(S_N \mid m_{udp}) = \frac{prob\ (mudp|SN)prob(SN)}{prob\ (\text{mudp}|SN)prob(SN) + prob\ (mtcp|wSH)prob(SH)}$$

$$\lambda = \frac{prob\ (mudp|SN)60\%}{prob\ (\text{mudp}|SN)60\% + prob\ (mtcp|wSH)prob(SH)}$$

$prob\ (mudp|SN)$=1, $prob\ (mtcp|wSH)$=0, λ=1, μ=0

Player 1 has violation motivation with signal m_{udp}, so it does not have separate equilibrium with signal (m_{udp}, m_{tcp}).

B) state (m_{tcp}, m_{udp})



Graph 3.2.2. State (m_{tcp}, m_{udp})

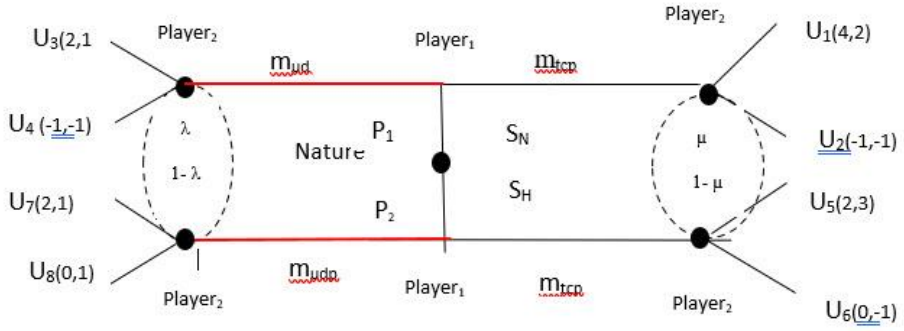
$$\text{Prob}(S_N \mid m_{tcp}) = \frac{prob\ (mtcp|SN)prob(SN)}{prob\ (\text{mtcp}|SN)prob(SN) + prob\ (mudp|wSH)prob(SH)}$$

$$\lambda = \frac{prob\ (mtcp|SN)60\%}{prob\ (\text{mtcp}|SN)60\% + prob\ (mudp|wSH)prob(SH)}$$

$prob\ (mudp|SN)$=0 ,$prob\ (mtcp|wSH)$=1, λ=1, μ=0

Player 1 doesn't have any motivation for violation. It has 4 on the left and 2 on the right. It doesn't have violation motivation on the low, either. After knowing m1 and m2, the answer payoff of the player 2 is (Y, Y). So, the equilibrium strategy is possible on the signal with (m_{tcp}, m_{udp}). It has a weak sequential and Nash equilibriums separately as follows:

((m_{tcp}, m_{udp}), (Y, Y); λ=1, μ=0))



Graph 3.2.3. State (m_{udp}, m_{udp})

C) state of (m_{udp}, m_{udp})

$$\text{Prob}(S_N \mid m_{tcp}) = \frac{prob\ (mtcp|SN)prob(SN)}{prob\ (\text{mtcp}|SN)prob(SN) + prob\ (mtcp|wSH)prob(SH)}$$

$$\lambda = \frac{prob\ (mudp|SN)60\%}{prob\ (\text{mudp}|SN)60\% + prob\ (mtcp|wSH)40\%}$$

$prob\ (\ mtcp|SN)$ =1, λ=1, $\Pi_2(Y \mid m_{udp}) = 1$, $\Pi_2(N \mid m_{udp}) = -0.2$

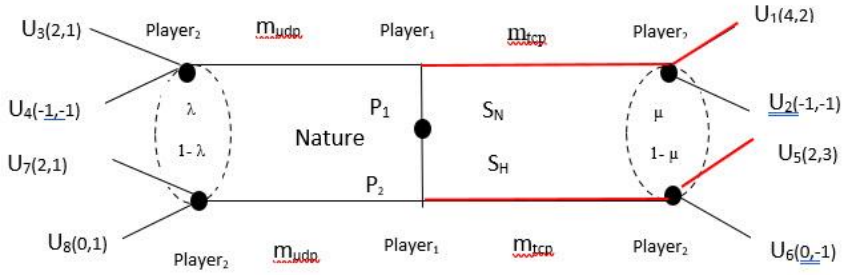
Therefore, the best work is Y.

$\Pi_2(Y \mid m_{tcp}) = 3-\mu$
$\Pi_2(N \mid m_{tcp}) = -1$ ⇒ $\mu \leq 4$

If μ≤ 4, motivation for violation will be destroyed, and it has a pooling equilibrium.

((m_{udp}, m_{udp}), (Y, Y); λ=1, μ≤4))

D) state (m_{tcp}, m_{tcp})



Graph 3.2.4. State (mtcp, mtcp)

$$\text{Prob}\ (S_N \mid m_{tcp}) = \frac{prob\ (\ mtcp|SN)prob\,(SN)}{prob\ (\text{mtcp}|SN)prob\,(SN)+ prob\ (\ mtcp|wSH)prob\,(SH)}$$

$$\mu = \frac{prob\ (\ mudp|SN)60\%}{prob\ (\text{mudp}|SN)60\%+ prob\ (\ mtcp|wSH)prob\,(SH)}$$

$prob\ (\ mtcp|SN)$=1

μ=1, $\Pi_2(Y \mid m_{tcp})$=2, $\Pi_2(N \mid m_{tcp})$= –1

Therefore, the best work is Y.

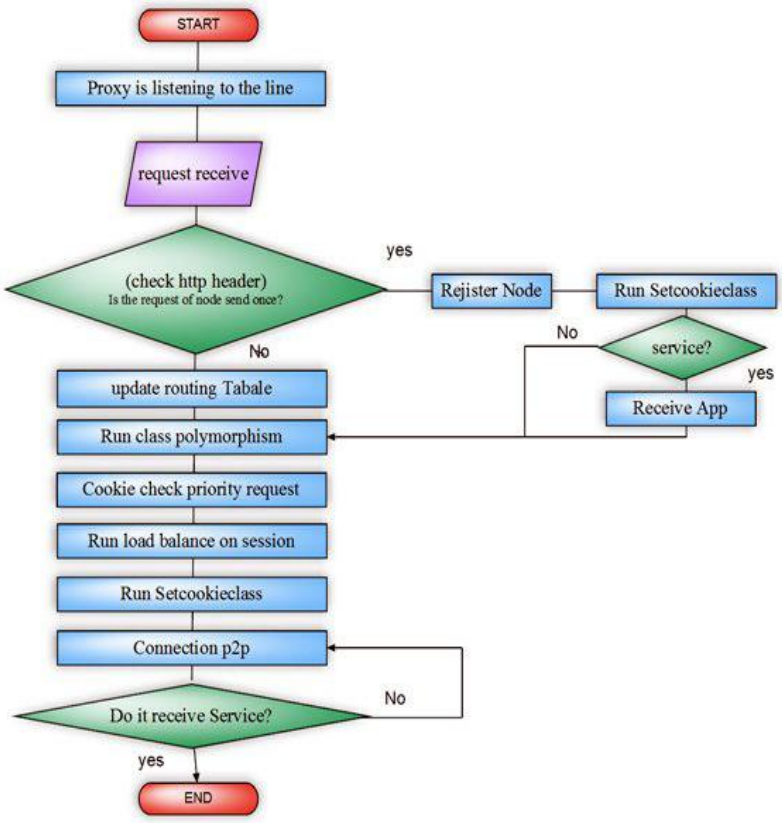
If μ≤ –1, motivation for violation will be destroyed, and it has a pooling equilibrium.

$((m_{tcp}, m_{tcp}), (Y, Y); \mu = 1, \lambda \leq -1))$ $\Pi_2(Y \mid m_{udp})=1$, $\Pi_2(N \mid m_{udp})=-\lambda$ ⇒ $\lambda \leq -1$

3.3. Algorithm of session-cookie stages in distributed cache

- Start.
- The proxy server always listens to the line.
- It checks the header of the transmitted request. If it is the first time that the node transmits the request, the node is registered, and the required application is received for service delivery to other nodes. Otherwise, it goes to the next step.
- The node's routing table is updated.
- Polymorphism class is executed based on the Session key and cookie pattern dynamically.
- The requests are prioritized.

- The session class is executed, and the load balancing is performed on the sessions.
- Cookie set class is executed and transmitted to the service applicant node.
- Peer-to-peer communication is performed between the client and the server.
- The nodes' connection is disconnected after the service reception.



Algorithm 1. Presenting and receiving service by the cluster's nodes

4. Experiments

4.1. Primary conditions

In this section, primary conditions for evaluation of the indicators are specified. At first, the characteristics of the nodes and the cookies are configured. The channel is wireless based on round-robin scheduling. In a wireless network, space is the communication channel. The wireless antenna has two most applicable propagation model of Free space and TwoRayGround for data transmission. Mostly the assumed radio propagation model is TwoRayGround. A network card is a wireless card of Wirelessphy. The type of queue interface of Ifq is DropTail/PriQueue. The type of queue is specified for data transmission from the nodes. In the queue, routing packets are preferred rather than data packets. The datalink layer is of LL type, and the MAC sub-layer is of 802.11 type. The type of antenna is Omni-directional, or one-directional. Omni-directional antenna propagates signal in all directions, and one-directional propagates the signal only in one specific direction. The queue length, Ifqn, is assumed 5MB that specifies at a time instant how many packets are waiting for transmission. If the count of the packets in the queue is more than the predefined count, it is removed automatically. The considered count of nodes is 12. A topography object is created that specifies the node is moving in a 10000×10000 space, and it is not going out of the space.

4.2. Experimental results

Figure 4.1. shows stored bandwidth efficiency in terms of the time of shared cache in milliseconds.

Based on Figure 4.1, the count of replied requests increases, so, QoE improves, and the proposed model has better quality.

Figure 4.2 shows using the proposed method, the increment of the count of the requests results in an increase in the average response time in milliseconds. It shows the distributed cache scalability in the proposed architecture. The distributed session based on the session and cookie technique by the increment of the count of the requests is more efficient.

Figure 4.3. depicts the hit rate of the cache in terms of the request's speed, the count of requests per second. The hit rate of the cache in the proposed architecture is higher.

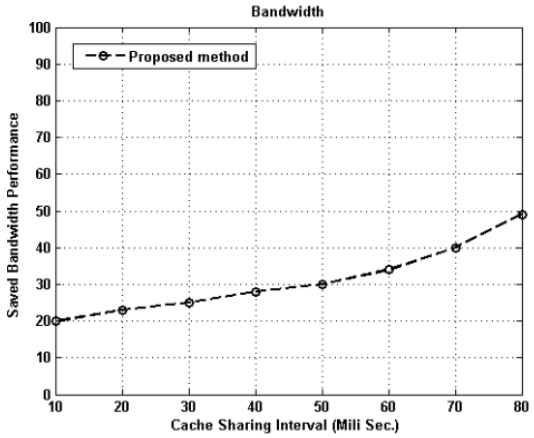


Figure 4.1. The consumed bandwidth

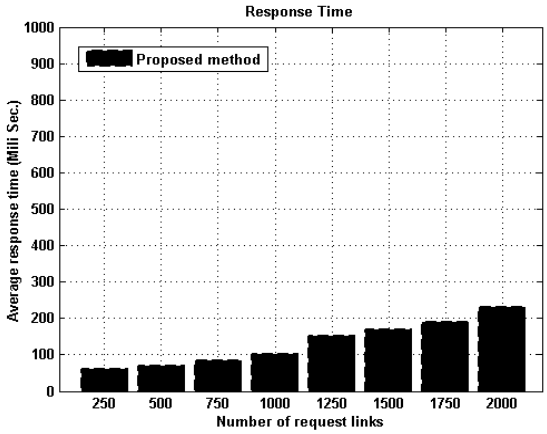


Figure 4.2. Average response time (ms)

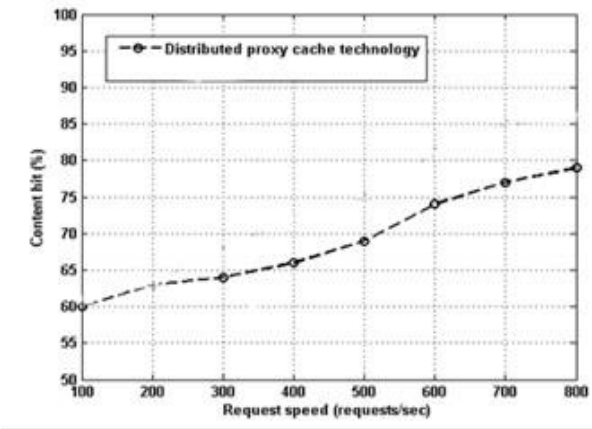


Figure 4.3. Hit rate of cache

5. Conclusion

In the proposed design, a distributed cache is designed as a site for serving the requested services using a proxy server and cookie capabilities, session, and polymorphism. The load balancing of round-robin is implemented in the sessions. The amount of hit rate of the cache, the response time of the service, and the consumed bandwidth are studied and calculated. The distributed cache hardware results in a single point of failure. If the hardware is defective, we are not worried about its failure and maintenance because it is transparent of place. So, it results in speed up and response time reduction. In response time, there is no single point of failure, and it is scalable and flexible, which is a great achievement in the proposed design. Also, based on games theory, the extended game model with incomplete information called weak sequential equilibrium is examined and proved that it has weak sequential equilibrium and Nash equilibrium in the mode ((mtcp, mudp), (Y,Y); λ=1, μ=0)). In the future work, the distributed cache architecture is implemented in the peer-to-peer system for serving automobiles and it is compared with the virtual cache creation method.

REFERENCES

1. Dias, R., Fiorese, A., Guardalben, L., Sargento, S. (2018). *A distributed caching architecture for Over-the-Top content distribution*. 2018 14th Annual Conference on Wireless On-demand Network Systems and Services (WONS), 2018, 9–16.
2. Dudykevych, V., Nechypor, V. (2016). *Detecting Third-Party User Trackers with Cookie Files.* 2016 Third International Scientific-Practical Conference Problems of Infocommunications Science and Technology (PIC S&T), 2016, 78–80.
3. Gibbons, R. (1992). *A Primer in Game Theory*. Pearson Higher Education, 1992.
4. Gonzalez, R., Jiang, L., Ahmed, M., Marciel, R., Cuevas, M., Metwalley, H., Niccolini, S. (2017). *The cookie recipe: Untangling the use of cookies in the wild*. Network Traffic Measurement and Analysis Conference (TMA), 2017, 1–9.
5. Haghi, N. (2016). *Model and algorithmsfor assignment and cache allocation problems in contents distributed networks*. University of Southampton faculty of social and human sciences school of Mathematics, 2016.
6. He, Hui, Cui, Lijie, Zhou, Fenglan, Wang Dong (2017). *Distributed proxy cache technology based on autonomic computing in smart cities*. Future Generation Computer Systems, 2017, vol:76, issue:0, 370-383.
7. Jiang, Zhihui, Ding, Zhiming, Gao, Xiaofeng, Chen, Guihai (2014). *DCP: An Efficient and Distributed Data Center Cache Protocol with Fat-Tree Topology*. The 16th Asia-Pacific Network Operations and Management Symposium, 2014.
8. Johnston, A. (2015). *Sip: Understanding the Session Initiation Protocol*. 2016 International Wireless Communications and Mobile Computing Conference (IWCMC), 2016, 363–368.
9. Johnson, T. A, Seeling, P. (2016). *On-device proxy and web cache for performance increases*. 2016, 13th IEEE Annual Consumer Comm. & Networking Conf. (CCNC), 2016, 268–269.
10. Jung, Kangsoo, Jo, Seongyong, Park, Seog (2015). *A game theoretic approach for collaborative caching techniques in privacy preserving location-based services*. 2015 International Conference on Big Data and Smart Computing (BIGCOMP), 2015, 59–62.
11. Li, Bo, Fu, Yanyan, Li, Zhihuai (2017). *The research and improvement of distributee caching system Memcached*. 2017 4th International Conference on Information, Cybernetics and Computational Social Systems (ICCSS), 2017, 460–463.
12. Li, Yiqiao, Wu, Yun, Duan, Xun (2017). Design and Implementation of Distributed Session Based on Beansdb, 2017 2nd IEEE International Conference on Computational Intelligence and Applications (ICCIA), 2017, 417–420.

13. Li, Yirui, Li, Zhuo (2014). A Resource Allocation Strategy for Multimedia Cloud Using Game Theory. 2014 IEEE Fourth Int. Conf. on Big Data and Cloud Computing, 2014, 222 - 226.
14. Manesh, T., Mohammed Sha, M., Vivekanandan, K. (2014). *Forensic investigation framework for P2P protocol*. Forensic investigation framework for P2P protocol, 2014, 256–264.
15. Mistry, Sujoy, Jaiswaly, Dibyanshu, Mukherjee, Arijit, Mukherjee (2015). P2P-Based Service Distribution over Distributed Resources. 2015 IEEE 29th International Conference on Advanced Information Networking and Applications, 2015, 515–520.
16. Moeini, Hessam, Yen, I-Ling, Bastani, Farokh (2017). *Efficient Caching for Peer-to-Peer Service Discovery in Internet of Things*. 2017 IEEE Int. Conf. on Web Services (ICWS), 2017, 196–203.
17. Mun, Ju Hyoung, Lim, Hyesook (2017). *Cache Sharing Using Bloom Filters in Named Data Networking*. Journal of Network and Computer Applications, 2017, vol. 90, 74–82.
18. Osborne, M. (2003). *An Introduction to Game Theory*. Oxford university press, 2003.
19. Osborne, M., Rubinstein, A. (1994). *A Course in Game Theory*. MIT press, 1994.
20. Renuka, B. S, Prafulla Shashikiran, G. T. (2020). Model of Load Distribution Between Web Proxy Servers Using Network Traffic Analysis. SN Computer Science, 2020.
21. Rimal, B. P., Maier, M. (2017). *Workflow Scheduling in Multi-Tenant Cloud Computing*. IEEE Transactions on Parallel and Distributed Systems, 2017, Vol. 28, Issue 1, 290–304.
22. Song, Mengjing, Li, Hui, Wu, Hao (2015). *A Decentralized Load Balancing Architecture for Cache System*. 2015 International Conference on Cyber-Enabled Distributed Computing and Knowledge Discovery, 2015, 114–119.
23. Song, Wei, Chen, Fangfei, Jacobsen, Hans-Arno, Xia, Xiaoxu, Ye, Chunyang, Ma, Xiaoxing (2017). *Scientific Workflow Mining in Clouds*, IEEE Transactions on Parallel and Distributed Systems, 2017, Vol. PP, Issue: 99, 290–304.
24. Sumalatha, M. R, Priyanka, N. Ahana (2016*). Load Balancing Using Improved Persistence by Cookie Insertion Method*. 2016 International Conference on Recent Trends in Information Technology (ICRTIT), 2016, 1–6.
25. Tanenbaum, A. (2017). *Distributed system principle and paradigms*. CreateSpace Independent Publishing Platform, 2017.
26. Wang, Tong, Wang,Yunfeng, Wang, Xibo, Cao Yue (2020). *A detailed review of D2D cache in helper selection*, 2020.
27. Xiang, Min, Jiang, Yuzhou, Xia ,Zhong, Huang, Chunmei (2020). *Consistent hashing with bounded loads and virtual nodes-based load*. In Cluster Computing, 2020.
28. Zhang, Yin, Li, Yujie, Wang, Ranran, M. Shamim, Hossain, Lu, Huimin (2020). *Multi-Aspect Aware Session-Based Recommendation for Intelligent Transportation Services*. IEEE Transactions on Intelligent Transportation Systems, 2020, 1–10.

*** * ***

Bita JAFARI is a PhD Student of Computer Systems Engineering at the Islamic Azad University, Rasht Branch, Iran. Her favorite topics include: Information technology, artificial intelligence, network architecture.

Peyman BAYAT is a member of the Computer Engineering Department at the Islamic Azad University, Rasht Branch, Iran. His academic rank is Assistant Professor. His favorite topics include: Information technology, artificial intelligence, network architecture.