International Journal of Innovative Technology and Exploring Engineering (IJITEE)
ISSN: 2278-3075 (Online), Volume-8 Issue-11, September 2019

Optimizing and Load Balancing for Flash Crowd to Improve Quality of Service in Content Delivery Network

Meenakshi Gupta, Atul Garg

Abstract: ***The online access has been increasing rapidly with the digitization of information, cheaper Internet service and affordable devices to access the Internet. This entails for not only handling increasing number of web requests but also meeting Quality of Service (QoS) requirements of end-users. Content Delivery Network (CDN) system is used to make better the performance of origin server by storing the popular contents on surrogate servers. The contents are disseminated to the web users through surrogate servers. The performance of CDN system relies on the selection of appropriate surrogate server to satisfy end-users' requests. The proposed method named Load Balancing using Neighbors and Utility Computing (LBNUC) takes into account requests arrival rate, load on surrogate servers, end-users' changing demand and capacity of surrogate servers. The aim is efficient utilization of CDN resources to minimize the time required to serve end-users requests and the cost of servicing requests. This method is also effective in handling of flash crowd situation by monitoring request rate. It handles this situation with support from neighbor surrogate servers and arranging additional resources, if required, through utility computing to meet QoS requirement of end-users.***

Keywords: Content delivery network, redirection, load balancing, flash crowd, utility computing, QoS.

I. INTRODUCTION

Content Delivery Network (CDN) Service providers have hundreds of thousands of surrogate servers deployed at various locations all over the world. It has been a trend to make use of CDN services by content providers of popular websites to deliver web contents to large number of geographically distributed end-users (clients) on behalf of them. The purpose is to provide better QoS of web content delivery to end-users resulting in enhanced reputation and increased revenues to content providers [1]. This requires replication of contents [2] on surrogate servers so that end-users' requests can be directly serviced through them without contacting origin server. The redirection technique also significantly affects the performance of CDN system. It requires the appropriate selection of surrogate server to satisfy users' requests otherwise it may result in improper utilization of CDN resources. This needs that redirection system has updated information regarding the status of surrogate servers. Further, some of the web contents go viral and become an online sensation within a short span of time.

Manuscript published on 30 September 2019.
*****Correspondence Author(s)
Meenakshi Gupta, Research Scholar, MMICT&BM (MCA), Maharishi Markandeshwar (Deemed to be University), Mullana, Haryana, India.
Atul Garg, Associate Professor, MMICT&BM (MCA), Maharishi Markandeshwar (Deemed to be University), Mullana, Haryana, India.

The reasons may be size of sharer's audience, information about some popular news or event etc. This leads to flash crowd situation in which request rate temporarily increases multifold for particular web contents. This may severely degrade the performance of the web servers servicing these web contents during that time span. It requires extra resources to handle it. However, content providers may not go for over provisioning to handle it as occurring of flash crowds is not so frequent. It also requires precise estimation of the resources required for handling it; otherwise it will lead to either poor web performance or higher financial cost.

This work suggests a hybrid technique for load balancing during normal traffic as well as flash crowd situation. It is adaptive in nature that prevents overloading of servers in both scenarios. It considers the efficient utilization of existing CDN resources and arranging extra resources temporarily as and when required using utility computing without going for over provisioning. The purpose is to enhance end-users experience and minimize the cost incurred by CDN service providers; ultimately by the content providers. The proposed technique LBNUC achieves better performance concerning average response time to service end-users requests.

II. RELATED WORK

Several techniques have been introduced to redirect end-users' requests to the most suitable surrogate server. These techniques can be categorized as non-adaptive or adaptive depending upon whether based on some heuristics or current state of the network and servers [3], [4]. Non-adaptive techniques are generally less complex and require less bandwidth, however, not as efficient as adaptive ones. Adaptive redirection techniques have been studied from various aspects such as dynamic replica placement [5], network distance proximity [6]-[8], bandwidth availability [9] and load on servers [8], [10]-[12].

Most of the studies considered jointly the problem of object replication and request routing [13]-[16] as request redirection decision is affected by the availability of requested contents on the servers. Redirection can be client-side [17] or server-side [18] based on the point where the redirection decision is made. The primary focus is to improve the response time [10], [11], [19] and minimize costs [20] to service requests from end-users. The redirection strategies can be implemented in distributed or centralized way, however, distributed strategy is more efficient as it is more scalable and robust [21]-[23].

Further, sometimes there is sudden multifold increase in request for some objects resulting in overloading or even crash of server. This situation termed as flash crowd is usually unpredictable and for short span of time. It deteriorates QoS for end-users severely for this duration. Several studies have been conducted to understand the nature and effects of flash crowd [24]-[26]. Long term provisioning to handle it is not economical due to only short term requirement of additional resources to handle it. Therefore, handling of this situation requires its detection and treatment as earlier as possible in order to keep the convergence time minimum [27], [28]. Several strategies have been suggested to manage it such as admission control [29]-[31], cooperation from users [32]-[34], P2P network [34], [35], cloud computing [36], [37] and cooperation between servers [38]. Some studies suggest proactive strategies [39], [40] to protect servers from excessively overloading during flash crowd. The precision in estimation of additional resources required is significant, otherwise either it can not be handled properly or will lead to over provisioning resulting in extra financial cost. As flash crowd situation is temporary and infrequent in nature, therefore, redirection mechanism needs to be adaptive according to current situation whether normal or flash crowd [41].

III. PROBLEM STATEMENT

In this work, it is considered that objects from origin server are replicated on surrogate servers following SCnP method [42]. The problem is to select suitable surrogate server for servicing the request from end-user. Whether the request should be serviced by the requested surrogate server or it should be redirected to another surrogate server/origin server. Further, how to handle the requests, if servers are overloaded or flash crowd occurs. The challenge is to satisfy end-users' requests with agreeable QoS and also make efficient and better utilization of existing CDN resources.

Based on the above consideration, the request redirection and load balancing problem is formulated as follows. The system consists of an origin server, K number of objects originally stored on origin server, N surrogate servers and a set of C clients associated with each surrogate server. The surrogate server $n \in \{1,2,\ldots N\}$ has SSZ_n bytes of storage capacity. The object $k \in \{1,2,\ldots K)$ has size OZ_k in bytes with probability p_k that a client requests the object. The objects from origin server are replicated on SS_n subject to storage capacity constraint using SCnP method.

Each surrogate server is aware of that which objects are replicated on neighbor surrogate servers associated with same CDN along with the information about request rate, service rate and load on the servers. Here, the load of a surrogate server is measured in terms of total number of requests pending in its queue. This information is shared among them at a regular interval. The SCnP method in [42] was evaluated using simulation consisting of one origin server, nine surrogate servers and nine set of clients. Firstly, it is evaluated that whether this method is effective even if the number of servers is decreased or increased. The simulation considers storage space used for replication on every surrogate servers is 30% of the total web contents size on the origin server and skewness α of contents popularity distribution is 0.7. The Table I shows that in every case, SCnP method for content replication gives the best performance in terms of average response time as compared to other methods.

Table-I: Performance evaluation of content replication strategies based on average response time and effect of number of surrogate servers

	6 Servers	9 Servers	12 Servers
Random	9.18	7.41	7.15
Round-robin	3.01	5.28	3.80
Popularity	6.40	6.20	5.87
SCnP	2.50	1.67	2.35

IV. LBNUC METHOD

In SCnP method, if the requested surrogate server does not have the requested object, then it simply examines its neighbor surrogate servers for that object. If any of these servers has the object, then the request is redirected to that server. Otherwise the request is redirected to the origin server. This work proposes an adaptive method LBNUC to assign request to the most suitable server keeping in view not only the availability of object but also the load on servers and dynamically replicating objects based on change in demand for the object. It is a hybrid method that can manage distribution of load among servers during normal and flash crowd situation. The aim is to improve client-perceived performance of servicing requested web contents. The conceptual model of LBNUC method is shown in Figure 1.

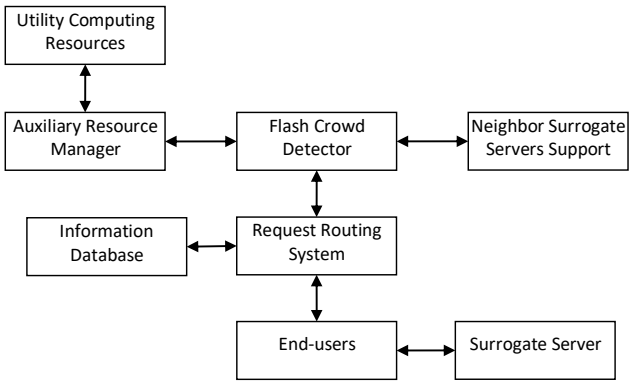


Fig. 1. Conceptual model of LBNUC method

When there is a request for an object, firstly it is redirected to nearby surrogate server. To decide a surrogate server for servicing a web request, the availability of requested object and load on the servers is checked. If the workload on requested surrogate server is within an acceptable limit, then no attempt is made for load distribution. If the load exceeds the specified limit, the load on its neighbor surrogate servers is checked and the request is redirected to the least loaded server keeping in view the servicing capacity of servers subject to the availability of the requested object on it. However, if the requested object is available only on the requested surrogate server, then the server is selected irrespective of the load on it.

Retrieval Number: K18680981119/19©BEIESP
DOI: 10.35940/ijitee.K1868.0981119
Journal Website: www.ijitee.org

Published By:
Blue Eyes Intelligence Engineering and Sciences Publication (BEIESP)

If none of the requested surrogate server and neighbor surrogate servers has the requested object, then it looks for origin server as all the objects are available on the origin server. If the number of requests redirected to origin server for the demanded object is less than a given threshold value, then the request is redirected to the origin server. Otherwise the least demanded object on the requested surrogate server is replaced with the requested object and the request is submitted to it.

LBNUC also manages to detect flash crowd effect at an early stage on the basis of request rate from end-users. As soon as the flash crowd effect is detected, the provision is made to ensure that the flash objects are available on the requested server. The information about flash objects is shared with neighbor surrogate servers and interval to provide the updated information to them is decreased. The neighbor servers propagate this information to their neighbors so that load can be distributed further. However, if the load is still not in controlled state, then utility computing is used to arrange additional resources and flash objects are replicated on these servers. The additional resources are allocated and released according to requirement that lead to manage the balance between performance and cost. The algorithm for LBNUC method is as follows:

Algorithm

Event 1: Server selection to service request

for each Q_k from C_n to SS_n
  if available(SS_n, O_k) then
    if satisfy_threshold_criteria (SS_n) then
      select SS_n to service Q_k and exit
for each NS_{ni} of SS_n
{
  if available(O_k) then
    find server with minimum weight
}
If found( )
  redirect Q_k to selected server and exit
if satisfy_load_criteria(OS, Q_k) then
  redirect Q_k to OS
else
{
  replace minimum serviced object from SS_n with O_k
  select SS_n to service Q_k and exit
}

Event 2: Flash detection

If overloaded(SS_n) then
{
  set multiple_ redirect on
  set reduced status update interval to neighbor servers
  find flash objects
  check flash objects availability on SS_n, if not available place on SS_n
}
else
{
  set multiple_redirect off
  set normal status update interval to neighbor servers
  set flash objects to NULL
}

Event 3: Flash information from neighbor surrogate servers

if (not flash(SS_n) and flash(NS_{ni})) then
  check flash objects on NS_{ni} availability on SS_n, if not available place on SS_n
else
{
  find O_{min}, minimum serviced object from SS_n
  find O_{max}, maximum redirected object to OS
  if $Q_{max} - Q_{min}$ > service rate of SS_n
    replace O_{min} on SS_n with O_{max}
}

Event 4: Employing utility computing resources

if overloaded(SS_n) then
{
  If additional_resources_required(SS_n) then
  {
    allocate required resources as neighbor servers of SS_n
    place flash objects on allocated resources
  }
  else
    release any additional resources allocated to SS_n
}

V. SIMULATION SETUP

LBNUC method is implemented using Network Simulator 2. The simulation setup has nine surrogate servers; each surrogate server is associated with a group of clients as shown in Figure 2. The simulation is carried out using library stated in [43]. The bandwidth of all links is same whereas arrival rate (λ) and service rate (μ) of the servers are different as given in Table II. Other parameters used for simulation are illustrated in Table III. Each surrogate server shares the information about its current status i.e. capacity, load and objects available on it with its neighbor surrogate servers on a regular interval. The popularity of objects is considered to follow zipf distribution.

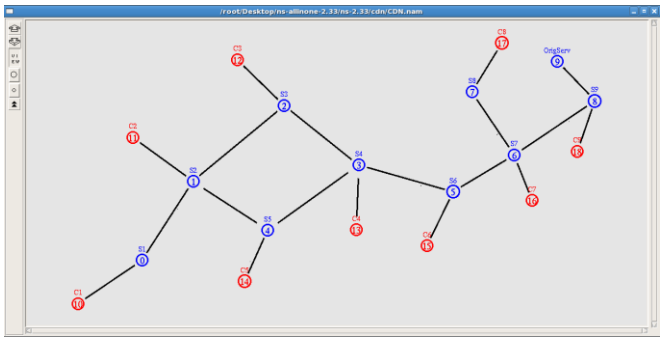


Fig. 2. Simulation topology

Table-II: Traffic characteristics

	1	2	3	4	5	6	7	8	9
λ_i[req/s]	12	10	7	10	13	11	11	14	17
μ_i[req/s]	25	26	28	34	20	22	22	14	20

Table-III: Simulation parameters

Parameters	Values
Bandwidth	10Mbps
Propagation delay	100msec
Update interval	10 sec
Sample time	1 sec
Object size	1000 bytes

Published By:
Blue Eyes Intelligence Engineering
and Sciences Publication (BEIESP)

Total objects on origin server	20
Objects replicated on every surrogate server	6 (30% of total objects)
Starting time of flash crowd	20th sec
Flash objects	2
Zipf skewness(α)	0.7
Simulation time	100 sec

VI. PERFORMANCE EVALUATION

In order to analyze LBNUC method for load balancing during normal condition and flash crowd, it is compared with random [44] and round-robin [45], non-adaptive methods; and least-loaded [46], adaptive method through simulation. The metrics used for evaluation are average response time for servicing web requests, queue length of pending requests on servers and percentage of requests completed to total number of requests. The different methods used for comparison are as follows:

Random: The requests from clients are assigned randomly among the requested surrogate server and its neighbor surrogate servers subject to availability of the requested object.

Round-robin: The requests are assigned among the requested surrogate server and its neighbor surrogate server son a round robin basis subject to availability of the requested object.

Least-loaded: The requests are assigned to the surrogate server that has the lowest load. In this case, the requested server has the information about the load on its neighbor surrogate servers and it is updated on a regular interval.

LBNUC: Clients' requests are sent to the nearest surrogate server. A request is satisfied by the server, if the requested object is available on it and the load criterion is satisfied. Otherwise, its neighbor surrogate servers are checked, if none of them is suitable, then it is redirected to the origin server. The origin server checks the number of requests redirected to it for the requested object. If it is not within limit, then the object least serviced by the requested server is replaced with the requested object and the request is submitted to the requested server.

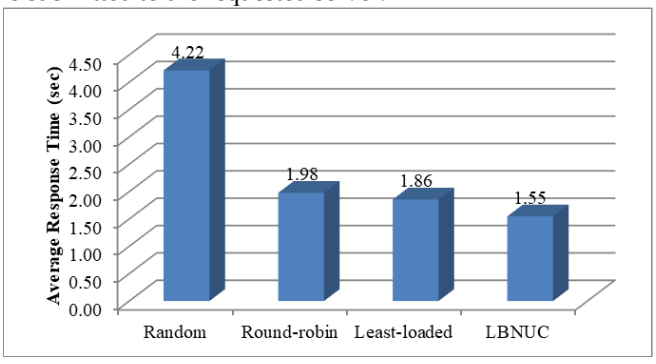


Fig. 3. Performance evaluation during normal load

Firstly, the performance under normal workload is measured to evaluate these strategies. Figure 3 illustrates the performance in terms of average response time for servicing web requests, when maximum storage space used for replication on any surrogate server is 30% of the total web contents size on origin server and skewness α of contents popularity distribution is 0.7. LBNUC method gives better response time to service the requests as compared to other methods.To simulate flash crowd scenario, the arrival rate at server 1, (λ_1) is increased from 12 requests per second to 50 requests per second (around 4 times) from time 20 sec onwards up to the completion of simulation i.e. 100 seconds and two of the objects are considered flash objects. To evaluate the effectiveness of different methods, the requests are generated from that client group for only flash objects. If the flash objects are not available on the requested surrogate server, then these objects are replicated on it by replacing with least demanded objects. By default, the requests can be redirected only once. However, during flash crowd, the requests can be redirected more than once so that the surrogate server getting the redirected request, if required, may further redirect it to distribute the load. The Figure 4 shows the performance of compared methods during flash crowd situation.

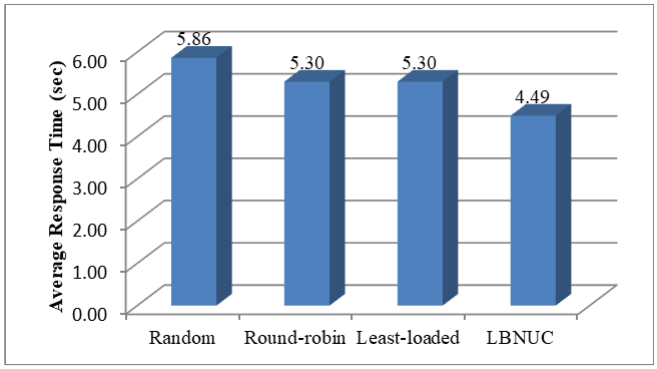


Fig. 4. Performance evaluation during flash crowd

The support from neighbor surrogate servers supplements to deal with the situation and combat the flash crowd effect. For this, the information about flash objects is provided by the requested server to its neighbor servers. The interval to update the neighbors is substantially reduced so that the neighbors can get the information about current status quickly. The neighbor servers check the availability of the flash objects on them, if not available, they replicate these objects using the same technique as used by the requested server. This leads to redirection of requests to neighbor servers and by the time the information about flash objects is further propagated to their neighbors. This helps to distribute the load among more and more surrogate servers to handle flash crowd situation. In addition to this, if redirection to origin server for any object is more than specified limit, then the object is replicated on the requested server to avoid overloading the origin server and the request is submitted to the requested server instead of redirecting it. Figure 5 shows that support from neighbor surrogate servers for load balancing during flash crowd yields further better performance.

Figure 6 shows the number of requests pending in queues of surrogate servers and origin server during flash crowd. This indicates that LBNUC method can effectively mitigate the flash crowd effect and achieve balancing the load distribution on servers, whereas, in other compared methods load on origin server continues to increase during flash crowd. This leads to increase in number of requests pending for getting response.

The results only from time 20sec to 40sec are highlighted for clarity of figures except for LBNUC method. This also affects unbalancing index that indicates balancing of load on servers. It is obtained by calculating the variance of the values of the queue length on the total number of the servers for every sample time and then averaging it on the number of samples. LBNUC method provides the best unbalancing index as well as the best performance in terms of percentage of requests completed as compared to total number of requests generated during simulation period. Table IV shows the comparative analysis of the methods during normal load and flash crowd.

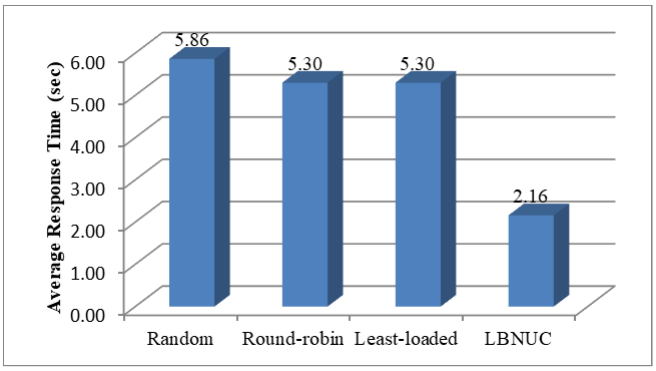
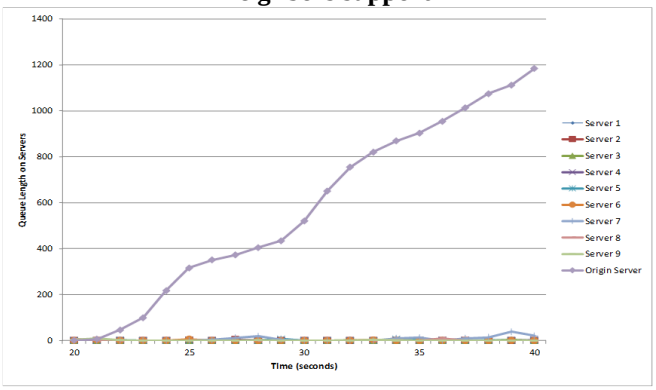
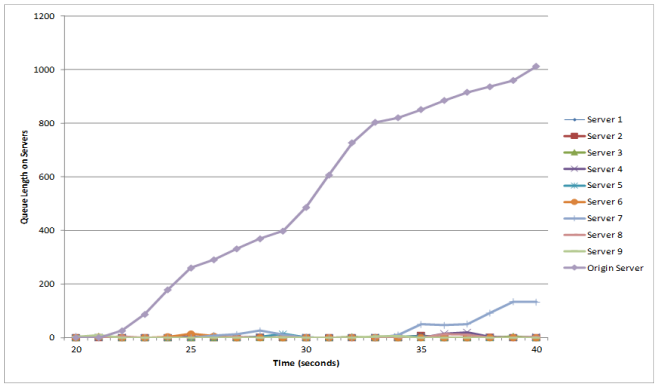


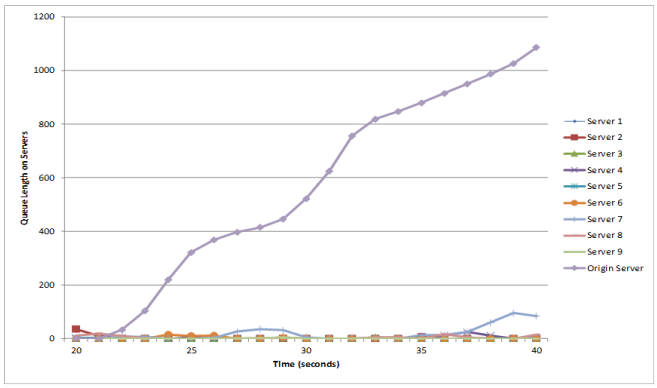
Fig. 5. Performance evaluation during flash crowd with neighbors support



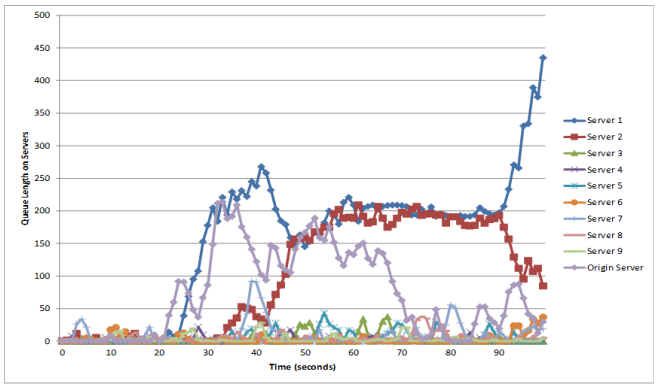
(a) Random



(b) Round-Robin



(c) Least-Loaded



(d) LBNUC

Fig. 6. Queue length on servers during flash crowd

Table-IV: Performance evaluation during normal load and flash crowd

	Normal Load			Flash Crowd		
	Requests Completed (%)	**Average Response Time (sec)**	**Unbalancing Index**	**Requests Completed (%)**	**Average Response Time (sec)**	**Unbalancing Index**
Random	83.12	4.22	185.10	64.29	5.86	595.82
Round-robin	96.37	1.98	45.27	71.07	5.30	473.21
Least-loaded	96.48	1.86	41.29	70.88	5.30	480.26
LBNUC	98.27	1.12	14.06	95.44	2.16	57.97

The existing number of surrogate servers may not be sufficient to maintain QoS provided to end-users as per service level agreement with content providers during flash crowd. For this some more server may be required to deal with the situation. As these resources are required temporarily, therefore, instead of making provision in existing infrastructure, utility computing resources may be used for this purpose as per the requirement. In this case, these extra resources may be released when not required. This helps CDN service providers to balance the cost and performance.

The results in Figure 7 show that LBNUC method outperforms other compared methods in terms of average response time by temporarily using one additional server to handle flash crowd.

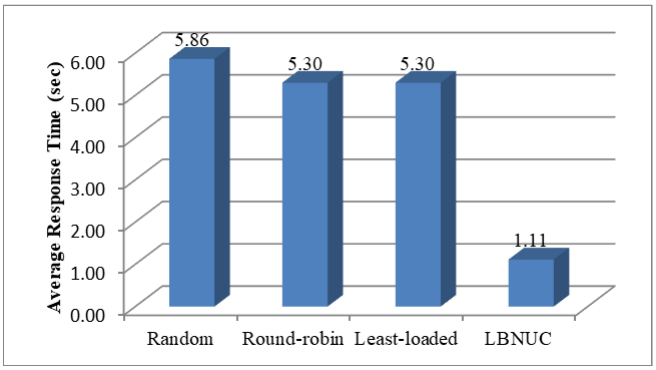


Fig. 7. Performance evaluation using utility computing resources

VII. CONCLUSION

The efficient utilization of CDN resources depends on allocating requests from end-users to appropriate surrogate servers. The proposed method LBNUC leads to efficient utilization of CDN resources by balancing the load on servers. It is effective in normal as well as flash crowd situation. During flash crowd, it detects it timely and takes the necessary action. The support from neighbor surrogate servers and using additional resources further facilitates to improve the performance. The results of the simulation clearly illustrate the effectiveness of LBNUC method in terms of user-perceived QoS and cost of servicing requests.

REFERENCES

1. M. Gupta, A. Garg, "CDN perspectives for quality delivery of contents," IOSR Journal of Computer Engineering, Special Issue - AETM'16, pp. 73-77, 2016.
2. M. Gupta, A. Garg, "A perusal of replication in content delivery network," Next-Generation Networks, Springer, Proceedings of CSI-2015, vol. 638, pp. 341-349, 2018
3. S. Sivasubramanian, M. Szymaniak, G. Pierre, and M. V. Steen, "Replication for web hosting systems," ACM Comput., vol. 36, no. 3, pp. 291–334, Sep. 2004.
4. A. K. Pathan and R. Buyya, "A taxonomy and survey of content delivery networks," Tech. Report, GRIDS-TR-2007-4, Grid Comput. Distrib. Syst. Lab. Univ. Melbourne, Aust., vol. 148, pp. 1–44, 2007.
5. F. Lo Presti, N. Bartolini, and C. Petrioli, "Dynamic replica placement and user request redirection in content delivery networks," Commun. 2005. ICC, vol. 3, pp. 1495-1501, 2005.
6. Z. M. Mao, C. D. Cranor, F. Douglis, M. Rabinovich, O. Spatscheck, and J. Wang, "A precise and efficient evaluation of the proximity between web clients and their local dns servers," in USENIX Annual Technical Conference, General Track, pp. 229-242, June 10, 2002.
7. T. Wang, J. Song, and M. Song, "A three-stage global optimization method for server selection in content delivery networks," Soft Comput., vol. 21, no. 2, pp. 467–475, 2017.
8. M. Pathan, C. Vecchiola, and R. Buyya, "Load and proximity aware request-redirection for dynamic load distribution in peering CDNs," in On Move to Meaningful Internet Sysyems, OTM Confederated Int. Conf., Springer, pp. 62–81, 2008.
9. S. Roy, D. Sarddar, and S. Roy, "Queueing based edge server selection in content delivery network using haversine distance," IJECCT, pp. 749-754, January, 2014.
10. T. Q. De Oliveira and M. P. Fernandez, "FuzzyCDN: Fuzzy redirection algorithm," Proc. - Int. Conf. Adv. Inf. Netw. Appl. AINA, pp. 437–444, 2013.
11. N. Ball and P. Pietzuch, "Distributed content delivery using load-aware network coordinates," Proc. 2008 ACM Conex. Conf. - Conex. '08, pp. 1–6, 2008.
12. P. Beniwal and A. Garg, "A comparative study of static and dynamic load balancing algorithms," International Journal of Advance Research in Computer Science and Management Studies, vol. 2, no. 12, pp. 1-7, 2014.
13. T. Bektaş, J.-F. Cordeau, E. Erkut, and G. Laporte, "Exact algorithms for the joint object placement and request routing problem in content distribution networks," Comput. Oper. Res., vol. 35, no. 12, pp. 3860–3884, Dec. 2008.
14. T. A. Neves, L. M. A. Drummond, L. S. Ochi, C. Albuquerque, and E. Uchoa, "Solving replica placement and request distribution in content distribution networks," Electron. Notes Discret. Math., vol. 36, pp. 89–96, 2010.
15. P. Amani, S. Bastani, and B. Landfeldt, "Towards optimal content replication and request routing in content delivery networks," IEEE Int. Conf. Commun., vol. 2015, pp. 5733–5739, 2015.
16. T. Neves, L. S. Ochi, and C. Albuquerque, "A new hybrid heuristic for replica placement and request distribution in content distribution networks," Optim. Lett., Jul. 2014.
17. A. Shaikh, R. Tewari, and M. Agrawal, "On the effectiveness of DNS-based server selection," in Proceedings IEEE INFOCOM 2001. Conference on Computer Communications. Twentieth Annual Joint Conference of the IEEE Computer and Communications Society (Cat. No.01CH37213), 2001, vol. 3, pp. 1801–1810.
18. S. Ranjan, R. Karrer, and E. Knightly, "Wide area redirection of dynamic content by internet data centers," IEEE Infocom 2004, vol. 2, no. 3, pp. 816–826, 2004.
19. Z.-M. Fei, S. Bhattacharjee, E. W. Zegura, and M. H. Ammar, "A novel server selection technique for improving the response time of a replicated service," in Proceedings. IEEE INFOCOM '98, the Conference on Computer Communications. Seventeenth Annual Joint Conference of the IEEE Computer and Communications Societies. Gateway to the 21st Century (Cat. No.98CH36169), 1998, vol. 2, p. 783–791.
20. W. Benchaita, S. Ghamri-Doudane, and S. Tixeuil, "On the optimization of request routing for content delivery," in ACM SIGCOMM Computer Communication Review, vol. 45, no. 4, pp. 347-348, Aug. 2015.
21. V. Cardellini, M. Colajanni, and P. S. Yu, "Request redirection algorithms for distributed web systems," IEEE Transactions on Parallel and Distributed Systems, vol. 14, no. 4, pp.355-368, 2003.
22. C. M. Chen, Y. Ling, M. Pang, W. Chen, S. Cai, Y. Suwa and O. Altintas, "Scalable request routing with next-neighbor load sharing in multi-server environments," 19th Int. Conf. Adv. Inf. Netw. Appl. Vol. 1 (AINA Pap., vol. 1, pp. 441–446, 2005.
23. S. Manfredi, F. Oliviero, and S. Pietro Romano, "Optimised balancing algorithm for content delivery networks," IET Commun., vol. 6, no. 7, pp. 733–739, 2012.
24. A. Dhingra and M. Sachdeva, "Recent flash events : A Study," Int. Conf. Commun. Comput. Syst., pp. 94–99, 2014.
25. A. Bhandari, A. L. Sangal, and K. Kumar, "Characterizing flash events and distributed denial-of-service attacks: An empirical investigation," Security and Communication Networks, vol. 9, no. 13, pp. 2222-2239, 2016.
26. J. Jung, B. Krishnamurthy, and M. Rabinovich, "Flash crowds and denial of service attacks- Characterization and implications for CDNs and web sites," in Proceedings of the 11th international conference on World Wide Web, ACM, 2002, pp. 293-304.
27. E. Lassettre, D. W. Coleman, Y. Diao, S. Froehlich, J. L. Hellerstein, L. Hsiung, T. Mummert, M. Raghavachari, G. Parker, L. Russell, M. Surendra, V. Tseng, N. Wadia, and P. Ye, "Dynamic surge protection: An approach to handling unexpected workload surges with resource actions that have lead times," in International Workshop on Distributed Systems: Operations and Management, Springer, Berlin, Heidelberg, pp. 82–92, 2003.
28. J. Coppens, T. Wauters, F. de Turck, B. Dhoedt, and P. Demeester, "Design and performance of a self-organizing adaptive content distribution network," 2006 IEEE/IFIP Netw. Oper. Manag. Symp. NOMS 2006, pp. 534–545, 2006.
29. X. Chen and J. Heidemann, "Flash crowd mitigation via adaptive admission control based on application-level observations," ACM Trans. Internet Technol., vol. 5, no. 3, pp. 532–569, 2005.
30. L. Xie, P. Smith, D. Hutchison, M. Banfield, H. Leopold, A. Jabbar, and J. P. G. Sterbenz, "From detection to remediation: A self-organized system for addressing flash crowd problems," IEEE Int. Conf. Commun., pp. 5809–5814, 2008.

31. X. Chen and J. Heidemann, "Experimental evaluation of an adaptive flash crowd protection system," ISI-TR-2003-573, vol. 2, no. July, 2003.
32. H. Nishiyama, H. Yamada, H. Yoshino, and N. Kato, "A cooperative user-system approach for optimizing performance in content distribution/delivery networks," IEEE J. Sel. Areas Commun., vol. 30, no. 2, pp. 476–483, 2012.
33. K. Yokota, T. Asaka, and T. Takahashi, "A load reduction system to mitigate flash crowds on web server," 2011 Tenth Int. Symp. Auton. Decentralized Syst., pp. 503–508, Mar. 2011.
34. C. Pan, M. Atajanov, and M. B. Hossain, "FCAN- flash crowds alleviation network using adaptive P2P overlay of cache proxies," no. 4, pp. 1119–1126, 2006.
35. A. Stavrou, D. Rubenstein, and S. Sahu, "A lightweight, robust P2P system to handle flash crowds," Proc. - Int. Conf. Netw. Protoc. ICNP, vol. X, pp. 226–235, 2002.
36. U. De Paula, D. De Oliveira, Y. Frota, and V. C. Barbosa, "Detecting and handling flash-crowd events on cloud environments," vol. V, 2015.
37. U. de P. Junior, L. M. A. Drummond, D. de Oliveira, Y. Frota, and V. C. Barbosa, "Handling flash-crowd events to improve the performance of web applications," Proc. – ACM Symp. on Appl. Comp., vol. 30, pp. 769-774, 2015.
38. S. Manfredi, F. Oliviero, and S. Pietro Romano, "A distributed control law for load balancing in content delivery networks," IEEE/ACM Trans. Netw., vol. 21, no. 1, pp. 55–68, Feb. 2013.
39. L. R. Moore, K. Bean, and T. Ellahi, "Transforming reactive auto-scaling into proactive auto-scaling," in Proceedings of the 3rd International Workshop on Cloud Data and Platforms, ACM, pp. 7–12, 2013.
40. D. Huang, M. Zhang, Y. Zheng, C. Chen, and Y. Huang, "Pre-allocation based flash crowd mitigation algorithm for large-scale content delivery system," Peer-to-Peer Netw. Appl., vol. 8, no. 3, pp. 493-500, Jun. 2014.
41. L. Wang, V. Pai, and L. Peterson, "The effectiveness of request redirection on CDN robustness," ACM SIGOPS Oper. Syst. Rev., vol. 36, no. SI, pp. 345-360, Dec. 2002.
42. M. Gupta and A. Garg, "Improving client-perceived performance based on efficient distribution of contents in content delivery network," Journal of Computational and Theoretical Nanoscience. Special issue on Big Data, Data Science &Analytics, IOT: Security Issues, Challenges, Concerns, to be published. 2019.
43. F. Cece, V. Formicola, F. Oliviero, and S. P. Romano, "An extended NS-2 for validation of load balancing algorithms in content delivery networks," SIMUTools 2010 - 3rd Int. ICST Conf. Simul. Tools Tech., p. 32, 2010.
44. R. Motwani and P. Raghavan, "Randomized algorithms." Chap- man & Hall/CRC, 2010.
45. Z. Xu and R. Huang, "Performance study of load balancing algorithms in distributed web server systems," CS213 Parallel and Distributed Processing Project Report, 1, 2009.
46. M. Dahlin, "Interpreting stale load information," IEEE Trans. Para. and Dist. Syst., vol. 11, no. 10, pp.1033-1047, 2000.

AUTHORS PROFILE

Meenakshi Gupta received degree of Master of Computer Applications from IGNOU, New Delhi and M. Phil. (Comp. Sc.) from Periyar University in 2005 and 2008 respectively. She is pursuing Ph.D. in Computer Applications from Maharishi Markandeshwar (Deemed to be University), Mullana (Ambala), Haryana, India. Currently, she is working as an Assistant Professor at Maharaja Agrasen Institute of Management and Technology, Jagadhri, Haryana. She has 13 years of teaching experience and has several national and international publications to her credit. She has life time membership of Computer Society of India. Her area of interest is fuzzy logic based systems and web optimization.

Atul Garg received degree of Master of Computer Applications from Kurukshetra University, Kurukshetra in 2004 and completed his Ph. D degree from Maharishi Markandeshwar University, Mullana (Ambala) in 2013. He has more than 15 years of teaching experience. Currently, he is working as an Associate Professor at M.M.I.C.T.&B.M., Maharishi Markandeshwar (Deemed to be University), Mullana (Ambala), Haryana, India. Two Ph. D. and M. Phil. scholars received their degree under his supervision and currently 5 scholars are ongoing. He has published two patents and more than 65 research articles in International / National Journals and Conferences. He has delivered number of expert talks. He is a Senior Member of the association of Universal Association of Computer & Electronics Engineers (UACEE), Australia, member in Society for Research and Development, member in the Institute for Computer Sciences, Social Informatics and Telecommunications Engineering (ICST), Belgium and Member in the International Association of Engineers, Hong Kong. His area of interest is web, Query Optimizations and Wireless networks.