International Journal of Engineering and Advanced Technology (IJEAT)
ISSN: 2249-8958 (Online), Volume-8 Issue-6S3, September 2019

Improving QoS by Enhancing Media Streaming Algorithm in Content Delivery Network

Atul Garg, Meenakshi Gupta

Abstract: ***Media streaming has gained popularity due to convenience of playing it at one's own leisure. It demands for smooth playing of media. However,with the increasing trend of media streaming and number of online users, it is getting difficult for content providers of popular media contents to handle media playing requests for popular media files. The number of simultaneous requests for media contents may affect uniform delivery of media contents and can lead to lower engagement of end-users.Content Delivery Network (CDN) plays an important role in streaming popular media contents by satisfying end-users' requests through surrogate servers. However, in order to enhance end-users experience, it is not sufficient to only reduce response time of media segments. It also requires to have lesser number of stalls during media streaming. This entails for redirecting requests to suitable surrogate servers as well as managingthetime duration between delivery of subsequent segments of a media file.The proposed methodnamed Stall Aware Media Streaming (SAMS) focuses on enhancing end-users experience by reducing wait time during media streaming.It keeps track of the possibility of stalls during media streaming and adjusts the media segments delivery rate to end-users accordingly. This results in meeting Quality of Service (QoS) requirement of end-users for media streaming in a better way by content providers.***

Keywords: Content delivery network, Media streaming, pre-roll buffer, stall rate, Quality of service

I. INTRODUCTION

Content Delivery Network has been widely used by popular servers to deliver contents to large number of geographically distributed users on the behalf of them. The web consists of several kinds of contents such as text, images, animation, audio and video. With the advancement in Internet and digital technology, web users prefer multimedia contents as compared to text and images. Online learning, games, movies, songs, live events, Internet TV, IPTV etc. all have contributed to the growth of multimedia contents. Further, with the cheaper Internet access, people are preferring media streaming rather than downloading media files. Many streaming services are available such as Hotstar, Netflix, Spuul, Voot, Hooq, Hungama, BoxTV etc. Some of these services are free and others charge at very nominal rates for getting the contents. The QoS requirements of media objects are different from traditional web objects such as html files, images. Further, streaming of media objects can't be handled efficiently using the same strategies as for downloading of the objects.

Manuscript published on 30 September 2019.
***** Correspondence Author (s)
Atul Garg, Chitkara University Institute of Engineering and Technology, Chitkara University, Punjab, India
Email: atul.garg@chitkara.edu.in, atulgarg1411@gmail.com
Meenakshi Gupta, Research Scholar, MMICT&BM (MCA), Maharishi Markandeshwar (Deemed to be University), Mullana, Ambala, Haryana, India. Email: mnkshgupta@gmail.com

It requires to consider that end-users (clients) do not face stalls during media streaming. If they have to stare at the rotating circle for too long while buffering during media play, they may leave even before a frame flashes for next segment of the media file. Therefore, the purpose of this work is to enhance user-perceived media streaming experience. This work suggests a method for reducing the wait time during media streaming whether this is wait time before starting the playing of media file or between playing of media file. The proposed method SAMS achieves better performance concerning waiting time during media streaming which in return leads to more user engagement.

II. RELATED WORK

The application of CDN system has been extended to multimedia streaming 1.The purpose is to provide better QoS of web content delivery to end-users resulting in enhanced reputation and increased revenues to content providers.Users that experience Quality of Experience (QoE) impairments such as lower bitrate, higher buffering and more fluctuations have lower engagement as compared to other users. Therefore, various techniques have been suggested to improve users' experience of media streaming.

Some of these techniques focus on selection of surrogate servers to replicate contents,and redirection, of users' requests to appropriate surrogate server in order to improve CDN system performance.Peer-to-peer network, clients' resources and cache proxies have been collaboratively used along with CDN resources to handle increasing demand for media streaming.

Other techniques suggest adaptive bit rate streaming for efficient utilization of available network bandwidth to provide high quality services to users as well as to maximize revenue to content providers and CDN service providers.Various video transcoding strategies have been used to generate different versions of a video and provide adaptive bitrates keeping in view users' QoS requirement. Basically, the purpose of all these techniques is to minimize stalls during media streaming and to improve end-users' perceived quality of service.

However, this work is based on improving client-perceived QoS of media streaming. The proposed method SAMS focuses on enhancing media streaming experience of clients while servicing all media requests with same bit rate.

III. PROBLEM STATEMENT

CDN system requires to efficiently replicate contents on surrogate servers and redirect clients' requests to suitable surrogate server.

As the size of media objects is multiple times more as compared to non-media objects, therefore, the methods to satisfy requests for non-media objects need to be changed accordingly. Considering the media objectssize and nature of media streaming, it is not efficient to deliver entire media object at once. Therefore, media streaming requires a different method for satisfying the media object request and it usually requires multiple responses against a request from clients.

Hence, a media object is partitioned into segments. Every segment has a playout duration. When there is a request from a client for a media object, it is deliveredin the form of multiple segments. However, all the segments of a media file are not delivered at once as severalmedia streaming requests are usually processed simultaneously. Therefore, the response time of segments of a media file may be different based on server and network condition. Considering this, some segments are pre-rolled in the buffer on client side before starting the playing of media file. For smooth playing, a client should just have the next segment(s) in the buffer before the playing of the last segmentin the buffer is finished.Otherwise, the clients will experience waiting in between the playing of a media file. Therefore, the challenge is to improve client perceived QoS of media streaming in terms of waiting time during pre-roll buffering as well as during playing of a media file.

Based on the above consideration, the media streaming problem is formulated as follows. The system consists of an origin server, K number of media 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 media object $k \in \{1,2,\ldots K)$ has size OZ_k in bytes with probability p_k that a client requests the object. The media objects from origin server are replicated on SSn subject to storage capacity constraint using Storage Cost and Performance based object placement (SCnP)method**Error! Reference source not found.**. Each media object is assumed to have samefixed encoding bit rate and same playing duration. The popularity of media objects is considered to follow zipf distribution.

Further, each surrogate server is aware of that which media 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 segment requests pending in its queue. This information is shared among them at a regular interval.

When there is a request from a client for a media object, it is redirected to suitable surrogate server. Instead of the complete media file, the request is inserted into the selected surrogate server queue only for the specified number of segments based on pre-roll buffer requirement. When the request for these segments is satisfied by the server, therequest for next segment of the media file is inserted in the server queue. The process for inserting subsequent segments of the media file is repeated until all the segments of the media are not sent to the client.

IV. SAMS METHOD

SAMS method is based on Load balancing using Neighbors and Utility Computing (LBNUC) method22 for selection of suitable server for servicing request for media object. In LBNUC method, the request for an object from client is redirected to suitable surrogate serverkeeping in view the availability of object, the load on servers and dynamically replicating objects based on change in demand for the object.This method mainly aims at minimizing the average response time of servicing web requests. However, this method is suitable fordownloading the objects and not for streaming media objects. In case of media streaming, the requirement is that clients should experience smooth playing of media files. Clients should not wait for a long time for starting of playing of media file and there should be minimum interruptions during playing of media file.

Therefore, SAMS method also keeps in view these considerations to improve QoS requirement of clients requesting media objects for streaming. On the one hand, it takes into account pre-roll time for buffering so that clients do not need to wait much for starting media file playing. On the other hand, it focuses on timely providing subsequent segments of the media file to minimize the number of stalls during its playing. When there is a request for a media object, it is redirected to nearby surrogate server. To decide a surrogate server for servicing a web request, firstly, the availability of the media object on the requested server is checked. Then, apart from current request, it checks the next segmentsto be inserted in the requested server queue for pending media files requests as well as new requests to be generated in that period, in order to reduce the interruptions during media streaming. If the time required to service these segments based on the service rate of the server is less than half of the initial pre-roll buffer time, then the request is submitted to the requested surrogate server. Otherwise load on its neighbor surrogate servers is checked, subject to the availability of the requested object on it.Here, it is not feasible to accurately calculate the next segments to be inserted in neighbor surrogate servers' queue, as this will increase the overhead in terms of passing more information to neighbor surrogate servers and increasing the frequency of updating this information. The request is redirected to the least loaded server keeping in view the servicing capacity of servers as well as delay on requested surrogate server for servicing subsequent segments of a media file. However, 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 requested 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. Once a request is submitted to a suitable server for servicing, initially some segments are inserted in the server queue based on initial pre-roll buffer condition.

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

At the client side, the media starts streaming after all the initial pre-roll segments are received. On the server-side, the requests for subsequent segmentsare inserted in the server queue one at a time.When the last segment for initial pre-roll buffer is sent to the client, the next segment is inserted in server queue until all the segments of the media file are processed. However, at this point of time it compares the time required to service requests pending in the server queue with playing time of initial pre-rolled segments. If the time required to service pending requests is more than that, it means playing of last segment in buffer will run out before receiving the next segment. In order to handle this situation, instead of one segment, consecutively two subsequent segments (if remaining) are inserted in server queue so that playing of the media object can be done without interruption further. Hence, SAMS method aims at improving clients' experience of media streaming by focusing on reducing pre-roll buffer time as well as stall rate during streaming of media object.

Algorithm

Event 1: Server selection to service media request

for each Q_k from C_n to SS_n
  calculate Qpndg, requests pending on SS_n
  calculate Qnew, estimation of new requests during that period
Qtot = Qpndg + Qnew
CalculateTtot, time required to service Qtot
  if available(SS_n, O_k) then
    if Ttot<= half of initial pre-roll buffer play time 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( ) and Ttot<=(initial pre-roll buffer play time+0.5) then
  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: Sending media segment(s) to client

for each seg(Q_k) in queueon SS_n
{
processseg(Q_k)
  if (seg(Q_k) >= initial pre-roll buffer size) then
  {
if (more segments for $Q_k$pending) then
{
insert request for next segment of $Q_k$in server queue
if (more segments for $Q_k$pending) then
if (serv_time(Qpndg)> initial pre-roll buffer play time) then
insert request for next segment of $Q_k$in server queue
  }
}
}

V. SIMULATION SETUP

Proposedmethod 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 1. The simulation is carried out using library stated in**Error! Reference source not found.**. The bandwidth of all links is same whereas arrival rate (λ) and service rate (μ) of the servers are different as given in Table I. Here the arrival and service rate are specified in terms of time interval. The arrival rate for a media object is considered in terms of a media file and not media segments. Other parameters used for simulation are illustrated in Table II. Each surrogate server shares the information about its current status i.e. capacity, load and media objects available on it with its neighbor surrogate servers on a regular interval. It is assumed that there is no pause/resume during play by clients.

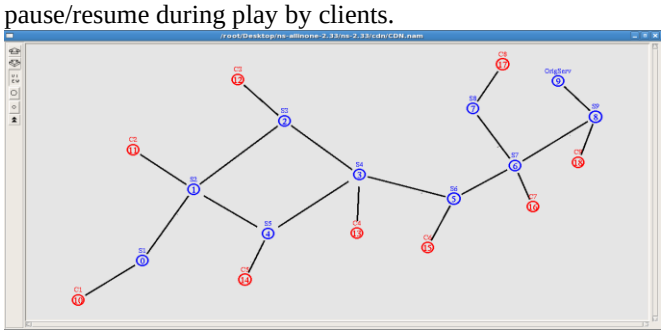

Fig. 1. Simulation topology

Table-I: Traffic characteristics

Time interval (sec)	Surrogate Server								
	1	2	3	4	5	6	7	8	9
λ_i	0.83	1	1	1	0.76	0.9	0.9	0.71	0.58
μ_i	0.040	0.038	0.035	0.029	0.05	0.045	0.045	0.07	0.05

Table-II: Simulation parameters

Parameters	Values
Total objects on origin server	20
Objects replicated on every surrogate server	6 (30% of total objects)
No. of segments in each file	10
Size of one segment	128KB
Media file size	1280KB
Segment play duration	1 sec
Initialpre-roll buffering size	3 segments
Link bandwidth	100Mbps
Propagation delay	100ms
Status update interval	10 sec
Zipf skewness(α)	0.7
Simulation time	100 sec

VI. PERFORMANCE EVALUATION

In order to analyze the proposed method for media streaming, it is compared with random**Error! Reference source not found.** and round-robin**Error! Reference source not found.**, non-adaptive methods; least-loaded**Error! Reference source not found.**adaptive method through simulation.

The metrics used for evaluation are average response time for servicing media segments, pre-roll buffering time, stalls during playing, percentage of requests completed to total number of requests in terms of receiving complete media file. 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 media object.

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

Least-loaded: The requests from clients are assigned to the surrogate server that has the lowest load subject to availability of the requested media object. 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.

SAMS: The decision for assigning requests to suitable surrogate server is taken not only on the basis of requests pending in server queue but also considering estimation of new segment requests to be generated. It also keeps track of possibility of stalls during media streaming and adjusts media segments delivery rate accordingly.

Figure 2 illustrates the performance in terms of delay for receiving the specified number of segments in pre-roll buffer on client side. The media streaming starts only after all the segments for pre-roll buffer are received. The results show that SAMS method performs better to other compared methods as clients have to wait less for starting media streaming. Figure 3 shows the interruptions during media streaming in terms of percentage of stalls. If the buffer on client side is exhausted before receiving the next segment, then the clients have to wait for receiving the next segment for playing. The proposed method outperforms other compared methods as clients experience lesser number of stalls during media streaming.The less waiting time during media streaming will help the media content providers to retain the existing clients as well as to increase the client base.Figure 4 shows that SAMS method is also performing better in terms of receiving complete media files.

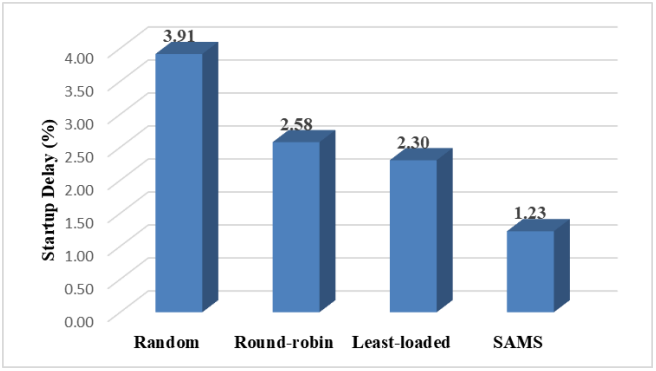

Fig. 2.Delay before starting media streaming (%)

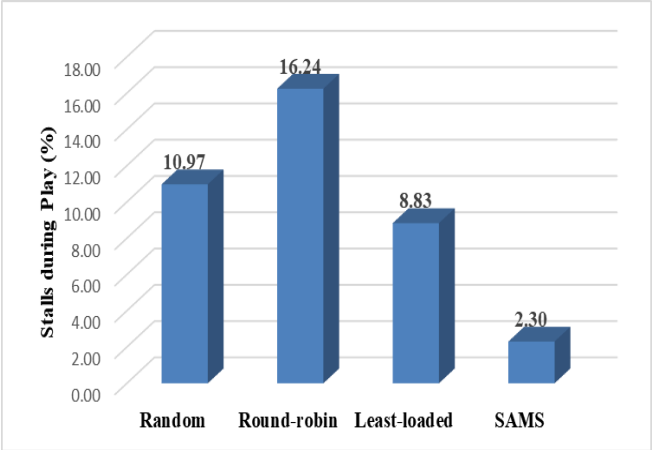

Fig. 3. Percentage of stalls during media playing

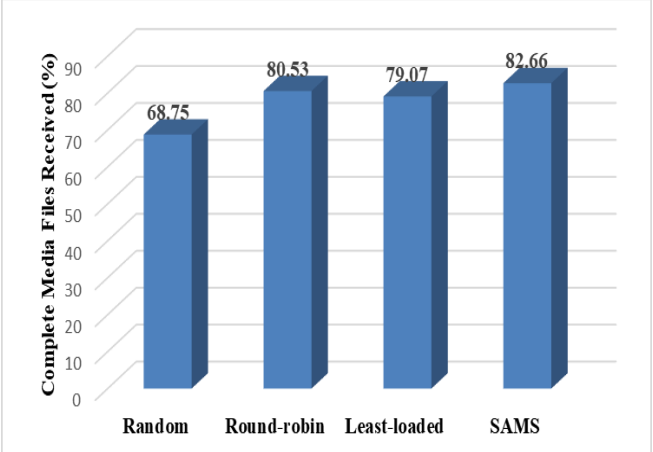

Fig. 4. Percentage of complete media files received to media files requested

Table IIIsummarizes the comparative analysis of the methods.Apart from time delay during media streaming, it shows that average response time to receive the segments is least in case of the proposed method. Here, unbalancing index 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. SAMS method provides the best unbalancing index representing the balancing of load on the surrogate servers as well as the origin server.

Table-III. Performance Evaluation

	Pre-roll Buffer Time (sec)	Stalls during Play (%)	Complete Media Files Received (%)	Average Response Time (sec)	Unbalancing Index
Random	3.91	10.97	68.75	1.45	106.30
Round-robin	2.58	16.24	80.53	1.26	56.61
Least-loaded	2.30	8.83	79.07	1.05	59.19
SAMS	1.23	2.30	82.66	0.50	11.54

VII. CONCLUSION

These days media streaming is preferred to downloading as the playback is instant. The users do not require towait for downloadingcomplete media file before playing it. This requires that playing of media file should not have much interruptions, otherwise, the users usually get annoyed and switch to another media file, may be on some other website. This work suggests SAMS method that leads to better online media streaming experience due to less fluctuation/buffering during media playing. Basically, the aim is to improve the performance of CDN system and as a result user-perceived web content delivery performance. This will result in more satisfaction to end-users and in return better reputation and profit to content providers and CDN service providers.

REFERENCES

1. B. Molina, J. Calvo, C. E. Palau, and M.Esteve, “Analyzing content delivery networks,” *Advanced Content Delivery, Streaming, and Cloud Services*, pp. 203-218, 2014.
2. M. Gupta, A. Garg, “CDN perspectives for quality delivery of contents," *IOSR Journal of Computer Engineering*, Special Issue - AETM'16, pp. 73-77, 2016.
3. L. Yala, P. A. Frangoudis, and A. Ksentini, “QoE-aware computing resource allocation for CDN-as-a-service provision,” *IEEE Global Communications Conf.*, pp. 1-6, Dec. 2016.
4. A. Ahmed, Z. Shafiq, and A. Khakpour, “QoEanalysis of a large-scale live video streaming event,”*ACM SIGMETRICS Performance Evaluation Review*, pp. 395-396, June 2016.
5. 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.
6. M. Yang and Z. Fei, “A model for replica placement in content distribution networks for multimedia applications,”*IEEE Int. Conf. Communications.*,vol. 1, pp. 557–561, 2003.
7. M. Vochin, E. Borcoci, S. G. Obreja, J. M. Batalla, and D. Negru, “Performance analysis of an adaptive media content streaming system,” *IEEE Int. Conf. Communications.*, pp. 149-152, June 2016.
8. J.B. Chen and S.-J. Liao, “A fuzzy-based decision approach for supporting multimedia content request routing in CDN,” *Int. Symp. Parallel Distrib. Process. with Appl.*, pp. 46–51, Sep. 2010.
9. J. Dai, Z. Hu, B. Li, J. Liu, and B. Li, “Collaborative hierarchical caching with dynamic request routing for massive content distribution,” *Proc. IEEE INFOCOM*, pp. 2444–2452, Mar. 2012.
10. H. Yin, C. Lin, Q. Zhang, Z. Chen, and D. Wu, “TrustStream: A secure and scalable architecture for large-scale internet media streaming,” *IEEE Trans. Circuits Syst. Video Technol.*, vol. 18, no. 12, pp. 1692–1702, Dec. 2008.
11. J. F. A. e Oliveira, Í. Cunha, E. Miguel, and S. V. A. Campos, “AERO: adaptive emergency request optimization in CDN-P2P live streaming,”*IEEE Global Communications Conf.*,pp. 1-7, Dec. 2017.
12. D. Xu, S. S. Kulkarni, C. Rosenberg, and H. Chai, “Analysis of a CDN-P2P hybrid architecture for cost-effective streaming media distribution,” *Multimedia Systems*,vol. 11, no. 4, pp. 383-399, 2006.
13. B. A. Abubakar, M. Petridis, D. S. Gill, and S. M. Gheytassi, “Adaptive CDN-based bandwidth conserving algorithm for mobile IPTV,” *8th Int. Conf. Adv. Comput. Intell.*, pp. 400–406, 2016.
14. Y.Miyauchi, N. Matsumoto, N. Yoshida, Y.Kamiya, and T.Shimokawa, “Adaptive content distribution network for live and on-demand streaming,” *ARCS Workshops*, pp. 27-37, 2012.
15. C. Liu, I. Bouazizi, M. M. Hannuksela, and M. Gabbouj, “Rate adaptation for dynamic adaptive streaming over HTTP in content distribution network,” *Signal Process. Image Commun.*, vol. 27, no. 4, pp. 288–311, 2012.
16. D. Zhang, H. He, and W. Li, “Bitrate allocation among multiple video streams to maximize profit in content delivery networks,” *Pers. Ubiquitous Comput.*, vol. 20, no. 3, pp. 385–396, 2016.
17. H.A. Nekrasov and A. Y. Romanov, “Distributed load balancing for the adaptive video streaming using CDN with ring system of server consolidation by circulant topology,” *Journal of Physics: Conference Series*, vol. 1210, no. 1, pp. 012105, Mar. 2019.
18. Z. Wang, L. Sun, C. Wu, W. Zhu, Q. Zhuang, and S. Yang, “A joint online transcoding and delivery approach for dynamic adaptive streaming,” *IEEE Trans. Multimed.*, vol. 17, no. 6, pp. 867–879, 2015.
19. M. S. Hossain and A. El Saddik, “QoS requirement in the multimedia transcoding service selection process,” *IEEE Trans. Instrum. Meas.*, vol. 59, no. 6, pp. 1498–1506, 2010.
20. A. Alabbasi, V. Aggarwal, T. Lan, Y. Xiang, M. Ra, and Y. R. Chen, “FastTrack : minimizing stalls for CDN-based over-the-top video streaming systems,”*IEEE Trans. Cloud Computing*, June 2019.
21. 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.
22. M. Gupta and A. Garg, “Optimizing and load Balancing for flash crowd to improve quality of service in content delivery network”, *International Journal of Innovative Technology and Exploring Engineering*,vol. 8, no. 11, pp. 2568-2574, Sep. 2019.
23. 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.
24. R. Motwani and P. Raghavan, “Randomized algorithms.” *Chap- man & Hall/CRC*, 2010.
25. 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.
26. M. Dahlin, “Interpreting stale load information,” *IEEE Trans. Para. and Dist. Syst.*, vol. 11, no. 10, pp.1033-1047, 2000.