JETIR.ORG ISSN: 2349-5162 | ESTD Year : 2014 | Monthly Issue
JOURNAL OF EMERGING TECHNOLOGIES AND INNOVATIVE RESEARCH (JETIR)
An International Scholarly Open Access, Peer-reviewed, Refereed Journal

Improvised Provision for Load Balancing in Content Delivery Networks

`

[1]Abhijeet Gajanan Gaikwad, [2]Professor Suvarna D. Pingle, [3]Dr. M. M. Dhobe
[1]Mtech Student, [2]Professor Dept CSE [3] HOD Dept CSE
[1][2][3] Computer Science & Engineering
[1][2][3]PES College of Engineering, Aurangabad

Abstract—
CDN technology is very important in today's technology world. This work focuses on the load balancing Algorithm and architecture of CDN technology describing in details the how the contents being distributed through surrogate servers across the world and how the load balancing algorithm works. This paper presents the Content Delivery Network (CDN) architecture for load balancing of information. Meaning everybody using smart phones, PCs, and the Internet have already been using CDN (Content Delivery Network) services, and will go on to use CDN services for their entire life. This paper presents how four load balancing algorithms affect the total execution time of the parallel job using Improvised Load Balancing Framework. We will summarize the existing literature and propose a system that enables applications with large processing power requirements to run on any number of nodes, in order to reduce their processing time.

Index Terms— CDN, Load balancing, auto tuned algorithm, proportional algorithm, manual algorithm, reinforcement learning algorithm.

I INTRODUCTION

Recently, Java has emerged as a language of choice for parallel programming. As one of the open source tools, Java arouse a lot of interest among professional software engineers. By using the JPPF, they can use things that enable easier parallel programming [3]. One of the most helpful tools is the load balancing algorithms. Usually, other frameworks that support parallel programming use only one algorithm to schedule jobs.

Every CDN node (also known as Edge Servers) caches the static content of a website just like the pictures, CSS/JS files and different structural parts. The bulk of Associate in Nursing end-user's page load time is spent on retrieving this content, and then it is smart to supply these "building blocks" of a website in as several server nodes as doable, distributed throughout the globe. It conjointly suggests that the quicker performance of a hosted web site and a much better security from hacker attacks.

This is as a result of CDNs maintain multiple Points of Presence, i.e. servers store copies of identical content and apply a mechanism that gives logs and data to the origin servers. As an alternative of a standard client-server communication, 2 communication flows area unit used between consumer and the surrogate server, so between the surrogate server and also the origin or central server.

CDNs not solely guarantee a quicker expertise to your users, however they conjointly facilitate to forestall website crashes within the event of traffic surges - CDNs facilitate to distribute information measure across multiple servers, rather than permitting one server to handle all traffic. Without CDN longer distance, increased latency and slower load times [1].

If a supplier establishes a content server in a very single physical location from which it disseminates information, services, and knowledge to all or any of its users, the single server is probably going to become over burdened and its links will simply be saturated. The speed at that users will access the location might become erratically over the maximum request rate the server and its links can tolerate. Since it's not possible, with this approach, to adapt to the exponential growth of the net traffic, the centralized model of content serving is inherently un-scalable, incapable of adapting and produces performance losses once traffic bursts occur. Though this results in the conclusion that a definite quantity of servers should be adopted, a cluster of servers (also named as server farm, that's a multi-server localized design) isn't essentially an answer nevertheless.

Several works have been done in this area that performed very well. In a work, an overlay Content Distribution Network (CDN) was planned which might able to sustain the period delivery of knowledge streams. A prophetical management theme was sculptured to boost utilization of resources and the effectiveness of the planned resolution was evaluated throughout multimedia system streaming and interactive grid information [2]. In another work, a data central networking was planned during which focus is shifted from the end-points within the network to the data objects themselves, with less care being placed on from wherever the data is fetched [3].

A Meta CDN system was developed in another significant work that exploits 'Storage Cloud' resources, making AN integrated overlay network that has an occasional price, high performance CDN for content creators. Meta CDN removes the quality of addressing multiple storage suppliers, by showing intelligence matching and putting users' content onto one or several storage suppliers supported their quality of service, coverage and budget preferences [4].

In a work in [5], they designed a CDN and studied regarding the problems concerning proxy server placement with an objective to produce its purchasers with the simplest offered performance whereas intense as very little resource as attainable. They applied genetic algorithmic program to resolve the server placement downside. A top quality of Service (QoS)-driven performance modeling approach for peering CDNs was bestowed so as to predict the user perceived performance [6].

II Related Work

Request routing in a CDN is usually concerned with the issue of properly distributing client requests in order to achieve load balancing among the servers involved in the distribution network. Several mechanisms have been proposed in the literature. They can usually be classified as either *static* or *dynamic*, depending on the policy adopted for server selection [15]. Static algorithms select a server without relying on any information about the status of the system at decision time. Static algorithms do not need any data retrieval mechanism in the system, which means no communication overhead is introduced. These algorithms definitely represent the fastest solution since they do not adopt any sophisticated selection process. However, they are not able to effectively face anomalous events like flash crowds.

Dynamic load-balancing strategies represent a valid alternative to static algorithms. Such approaches make use of information coming either from the network or from the servers in order to improve the request assignment process. The selection of the appropriate server is done through a collection and subsequent analysis of several parameters extracted from the network elements. Hence, a data exchange process among the servers is needed, which unavoidably incurs in a communication overhead.

The redirection mechanisms can be implemented either in a *centralized* or in a *distributed* way [15]. In the former, a centralized element, usually called *dispatcher*, intercepts all the requests generated into a well-known domain, for example an autonomous system, and redirects them to the appropriate server into the network by means of either a static or a dynamic algorithm. Such an approach is usually adopted by commercial CDN solutions. With a distributed redirection mechanism, instead any server receiving a request can either serve it or redistribute it to another server based on an appropriate (static or dynamic) load-balancing solution.

Depending on how the scheduler interacts with the other components of the node, it is possible to classify the balancing algorithms in three fundamental models. [16]

In a rate-adjustment model, instead the scheduler is located just before the local queue: Upon arrival of a new request, the scheduler decides whether to assign it to the local queue or send it to a remote server. Once a request is assigned to a local queue, no remote rescheduling is allowed. Such a strategy usually balances the request rate arriving at every node independently from the current state of the queue. No periodical information exchange, indeed, is requested. In a hybrid-adjustment strategy for load balancing, the scheduler is allowed to control both the incoming request rate at a node and the local queue length. Such an approach allows to have a more efficient load balancing in a very dynamic scenario, but at the same time it requires a more complex algorithm.

In the context of a hybrid-adjustment mechanism, the queue-adjustment and the rate-adjustment might be considered respectively as a fine-grained and a coarse-grained process. Both centralized and distributed solutions present pros and cons depending on the considered scenario and the specific performance parameters evaluated. As stated in [17], although in some cases the centralized solution achieves lower response time, a fully distributed mechanism is much more scalable. It is also robust in case of dispatcher fault, as well as easier to implement.

Table 1.0 Literature Review Summarization

Author	Paper Title	Description
[1] M. Kyryk, N. Pleskanka	Adaptive Edge Compute Module in CDN Networks	The study explores the methods and principles of building content delivery networks and proposes an Edge Compute module to enhance service quality. A simulation modeling of the module, based on a developed algorithm, was conducted, revealing graphical dependencies on efficiency based on load and client requests. The results confirm the module's effectiveness in increasing requests.
[2] H. Nishiyama, H. Yamada	A Cooperative User-System Approach for Optimizing Performance in Content Distribution/Delivery Networks	The increasing demand for content delivery in wired/wireless heterogeneous networks is causing CDNs to degrade due to changes in user demand and wireless mobility. To address this, a cooperative server selection scheme is developed to maximize robustness to changes in traffic. This scheme is evaluated through extensive computer simulations, demonstrating that it effectively makes the content delivery system resilient against request fluctuations while minimizing system overloading. This approach is crucial for managing the increasing demand in heterogeneous networks.
[3] V. K. Adhikari et al	Measurement Study of Netflix, Hulu, and a Tale of Three CDNs	Netflix and Hulu are leading OTT content service providers in the US and Canada, with Netflix accounting for 29.7% of peak downstream traffic in 2011. Understanding their system architectures and performance can help improve future systems. Both platforms rely heavily on third-party infrastructures, with Netflix migrating most functions to the Amazon cloud and Hulu hosting services out of Akamai. They employ the same content distribution networks (CDNs) for video content delivery. However, both platforms assign CDNs without considering network conditions or optimizing user-perceived video quality. Performance measurements show significant variation in available bandwidths over time and geographic locations. A measurement-based adaptive CDN selection strategy and multiple-CDN-based video delivery strategy can significantly increase users' average available bandwidth.
[4] A. Saengarunwong et. al	A Two-Step Server Selection in Hybrid CDN-P2P Mesh-based for Video-on-Demand Streaming	Video content dominates internet traffic, and Video-on-Demand (VoD) can lead to delays and bandwidth consumption. Mesh-based Peer-to-Peer (P2P) video streaming is used in many applications due to its robustness against peer churn. A hybrid CDN-P2P is implemented, addressing issues with server selection and neighbor selections. A two-step server selection algorithm is proposed for CDN-P2P VoD streaming. The algorithm

		selects the closest server based on hop count and the server hosting a group of peers closest to the incoming client. The hybrid CDN-P2P overlay is both closest and consists of nearest neighbors. The performance of the algorithm is evaluated in terms of startup delay, discontinuity, distortion, server direct connections, server data bandwidth consumption, and peer data bandwidth contribution.
[5] J. Liu, Q. Yang et. al	Congestion Avoidance and Load Balancing in Content Placement and Request Redirection for Mobile CDN	Mobile network operators are increasingly integrating content delivery network (CDN) functionalities into their networks to improve their capacity for content-oriented services. A mobile CDN system with Base Stations (BSs) is considered, where they cooperate in replying user requests through backhaul links. However, blindly redirecting user requests can cause traffic congestion, necessitating congestion avoidance and load balancing. The study investigates the joint optimization problem of content placement and request redirection for BS-based mobile CDNs, focusing on network congestion and load balancing. The researchers use a stochastic optimization model and Lyapunov optimization technique to solve the long-term problem and develop an online algorithm for efficient content placement and request redirection. The algorithm's performance on optimality and network stability is evaluated, confirming its ability to achieve low transmission costs while avoiding congestion and balancing traffic loads.
[6] W. Yang, Y. Hu, L. Ding et.al	Viewer-Oriented CDN Scheduling on Crowdsourced Live Video Stream	Crowdsourced live video platforms like Douyu and Twitch distribute streams based on popularity, without considering viewer engagement. This paper analyzes Douyu and finds a gap between popularity-based stream distribution and platform profit. To fill this gap, a viewer-oriented algorithm is proposed to distribute streams and determine viewer's streaming source based on viewer engagement, aiming to maximize platform profit. This viewer-oriented approach will improve the Quality of Experience for viewers worldwide.
[7] H. -A. Tran, S. Souihi, D. Tran and A. Mellouk	MABRESE: A New Server Selection Method for Smart SDN-Based CDN Architecture	The text proposes integrating a software-defined network (SDN) into a content delivery network (CDN) architecture to provide a global view of the network and optimize server selection. The goal is to decouple the control plane from the forwarding plane for flexibility and programmability. The authors also develop a novel server selection algorithm based on the multi-armed bandit

		problem, which optimizes server selection function and provides good experimental results in terms of average response time and reward score.
[8] R. Torres, A. Finamore, J. R. Kim, M. Mellia, M. M. Munafo and S. Rao	Dissecting Video Server Selection Strategies in the YouTube CDN	This study examines the YouTube CDN to understand the mechanisms and policies used to determine video download data centers. Data is collected from five networks, including university campuses and ISP networks, in three countries. The study uses delay-based geolocation techniques to locate YouTube servers. The analysis is unique as it considers related YouTube flows, allowing for inferring key system design aspects. Results show that RTT between users and data centers influences video server selection, but other factors like load-balancing, diurnal effects, DNS server variations, and popular video content hotspots also influence selection.
[9] Zhou Wang, A. C. Bovik	Image quality assessment: from error visibility to structural similarity	The study introduces a new framework for assessing perceptual image quality, based on the degradation of structural information. It proposes a structural similarity index, which is demonstrated through intuitive examples and comparison to subjective ratings and state-of-the-art objective methods on a database of compressed images compressed with JPEG and JPEG2000. This approach complements traditional objective methods for assessing image quality.

III Proposed Methodology

Algorithms Used For Load Balancing

In the following, we will describe the most common algorithms used for the purpose of load balancing in a CDN. Such algorithms will be considered as benchmarks for the evaluation of the solution we have presented in this paper.

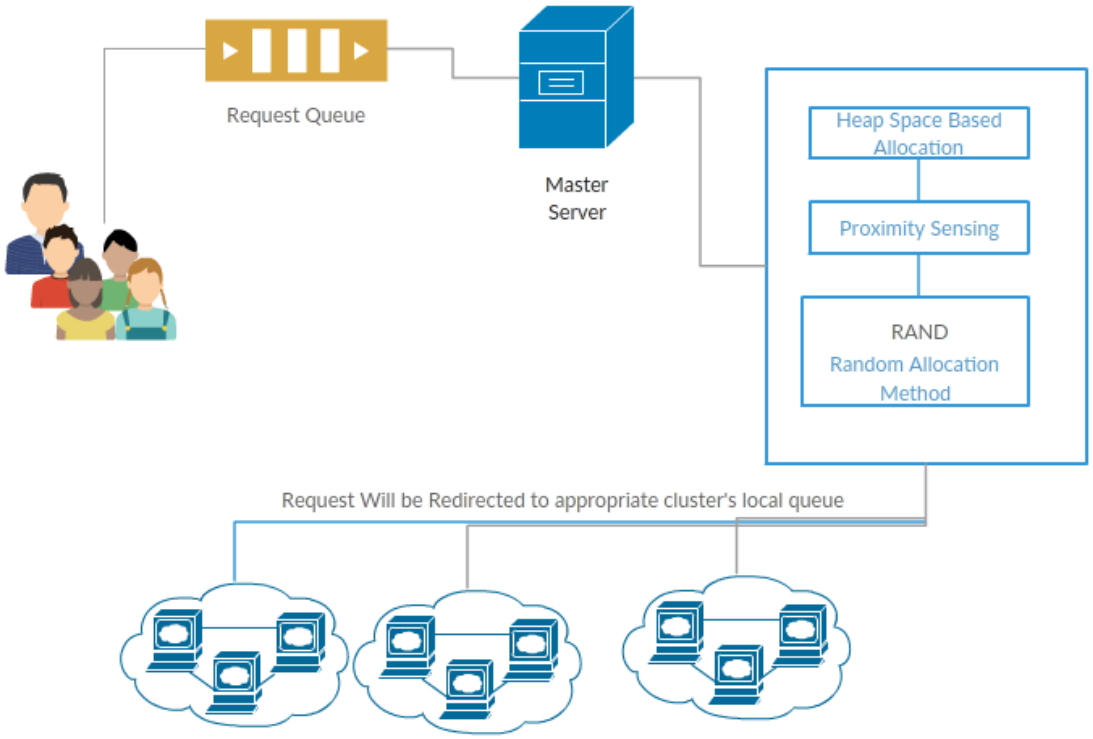

Figure 1.0 Proposed Architecture

The simplest static algorithm is the *Random b*alancing mechanism (RAND). In such a policy, the incoming requests are mainly distributed to the servers in the network with a uniform probability.

The *Least-Loaded* algorithm (LL) is a renowned dynamic strategy for load balancing. It assigns the incoming client request to the currently least loaded server. Such kind of approach is adopted in several commercial solutions. Unfortunately, it tends to rapidly saturate the least loaded server until the new message is propagated [12]. Alternative solutions can rely on *Response Time* to select the server: The request is allocated to the server that shows the fastest response time [13].

The *Two Random Choices* algorithm [14] (2RC) randomly chooses two servers and assigns the request to the least loaded one between them.

We propose a highly dynamic distributed strategy based on the periodical exchange of information about the status of the nodes in terms of load. By exploiting the multiple redirection mechanism offered by HTTP, our algorithm tries to achieve the global balancing

through a local request redistribution process. Upon arrival of a new request, indeed, the CDN server can either elaborate locally the request or redirect it to other servers according to a certain decision rule, which is based on the state information exchanged by the servers. Such an approach limits state exchanging overhead to just local servers. .

IV Conclusion

The above comparison shows that static load balancing algorithms are more efficient compared to adaptive ones and it is also ease to predict the behaviour of static algorithms. Adaptive algorithms using the variable granularity of work creates more network communication between servers and nodes. Increased network communication leads to increased total execution time of parallel program, and poor granularity of work can also significantly increase the total executing time.

V Acknowledgment

First and foremost, I would like to thank my guide, for unconditional guidance and support. I will forever remain grateful for the constant support and guidance extended by guide, in making this paper. Through our many discussions, he helped me to form and solidify ideas.

VI References

[1] M. Kyryk, N. Pleskanka and M. Pleskanka, "Adaptive Edge Compute Module in CDN Networks," 2023 IEEE 5th International Conference on Advanced Information and Communication Technologies (AICT), Lviv, Ukraine, 2023, pp. 41-43, doi: 10.1109/AICT61584.2023.10452684

[2] H. Nishiyama, H. Yamada, H. Yoshino and N. Kato, "A Cooperative User-System Approach for Optimizing Performance in Content Distribution/Delivery Networks," in IEEE Journal on Selected Areas in Communications, vol. 30, no. 2, pp. 476-483, February 2012, doi: 10.1109/JSAC.2012.120228.

[3] V. K. Adhikari et al., "Measurement Study of Netflix, Hulu, and a Tale of Three CDNs," in IEEE/ACM Transactions on Networking, vol. 23, no. 6, pp. 1984-1997, Dec. 2015, doi: 10.1109/TNET.2014.2354262.

[4] A. Saengarunwong and T. Sanguankotchakorn, "A Two-Step Server Selection in Hybrid CDN-P2P Mesh-based for Video-on-Demand Streaming," 2018 International Conference on Information and Communication Technology Convergence (ICTC), Jeju, Korea (South), 2018, pp. 499-504, doi: 10.1109/ICTC.2018.8539453.

[5] J. Liu, Q. Yang and G. Simon, "Congestion Avoidance and Load Balancing in Content Placement and Request Redirection for Mobile CDN," in IEEE/ACM Transactions on Networking, vol. 26, no. 2, pp. 851-863, April 2018, doi: 10.1109/TNET.2018.2804979.

[6] W. Yang, Y. Hu, L. Ding and Y. Tian, "Viewer-Oriented CDN Scheduling on Crowdsourced Live Video Stream," 2019 IEEE 2nd International Conference on Electronics and Communication Engineering (ICECE), Xi'an, China, 2019, pp. 112-117, doi: 10.1109/ICECE48499.2019.9058519.

[7] H. -A. Tran, S. Souihi, D. Tran and A. Mellouk, "MABRESE: A New Server Selection Method for Smart SDN-Based CDN Architecture," in IEEE Communications Letters, vol. 23, no. 6, pp. 1012-1015, June 2019, doi: 10.1109/LCOMM.2019.2907948.

[8] R. Torres, A. Finamore, J. R. Kim, M. Mellia, M. M. Munafo and S. Rao, "Dissecting Video Server Selection Strategies in the YouTube CDN," 2011 31st International Conference on

Distributed Computing Systems, Minneapolis, MN, USA, 2011, pp. 248-257, doi: 10.1109/ICDCS.2011.43.

[9] Zhou Wang, A. C. Bovik, H. R. Sheikh and E. P. Simoncelli, "Image quality assessment: from error visibility to structural similarity," in IEEE Transactions on Image Processing, vol. 13, no. 4, pp. 600-612, April 2004, doi: 10.1109/TIP.2003.819861.

[10] Matthew L., Content Delivery Network (CDN) 2A Reference Guide, detail, April 10, 2019, http://Informationweek.com/detail/RES/9959901 81212.html.

[11] OLNN.CN.CDN, The Content distributed network technology, server, May 28, 2019, http://olnn.cn/html/server/6/20070528/3660.html

[12] W. D. Zhao, Content routing algorithms achieving network proximity in content delivery networks, Journal of Zhejiang University, Vol. 38, No. 4, p.414 -419, April 2019.

[13] (2020) Reinforcement Learning: An Introduction - Richard S. Sutton and Andrew G. Barto (Book)

[14] Derek L. Eager, Edward D. Lazowska , John Zahorjan, "Adaptive load sharing in homogeneous distributed systems", IEEE Transactions on Software Engineering, v.12 n.5, p.662-675, May 2020 Monte Carlo method [Online]. Available

[15] M. Dahlin, "Interpreting stale load information," IEEE Trans. Parallel Distrib. Syst., vol. 11, no. 10, pp. 1033–1047, Oct. 2021

[16] R. L. Carter and M. E. Crovella, "Server selection using dynamic path characterization in wide-area networks," in Proc. IEEE INFOCOM, Apr. 2021, vol. 3, pp. 1014–1021.

[17] M. D. Mitzenmacher, "The power of two choices in randomized load balancing," IEEE Trans. Parallel Distrib. Syst., vol. 12, no. 10, pp.1094–1104, Oct. 2021.

[18] V. Cardellini, E. Casalicchio, M. Colajanni, and P. S. Yu, "The state of the art in locally distributedWeb-server systems," Comput. Surveys,vol. 34, no. 2, pp. 263–311, Jun. 2022.

[19] Z. Zeng and B. Veeravalli, "Design and performance evaluation of queue-and-rate-adjustment dynamic load balancing policies for distributed networks," IEEE Trans. Comput., vol. 55, no. 11, pp. 1410–1422, Nov. 2022.

[20] V. Cardellini, M. Colajanni, and P. S. Yu, "Request redirection algorithms for distributed Web systems," IEEE Trans. Parallel Distrib. Syst., vol. 14, no. 4, pp. 355–368, Apr. 2023.