Load Balancing in Cloud-based Content Delivery Networks using Adaptive Server Activation/Deactivation

Maggie Mashaly
Faculty of Information Engineering and Technology
German University in Cairo (GUC), Cairo/Egypt
Paul J. Kühn
Institute of Communication Networks and Computer Engineering
University of Stuttgart, Stuttgart/Germany
{maggie.ezzat@guc.edu.eg, paul.j.kuehn@ikr.uni-stuttgart.de}

Abstract. Content delivery networks have been widely used for many years providing service for millions of users. Lately, many of these networks are migrating to the cloud for its numerous advantages such as lower costs, increased performance, availability and flexibility. This work introduces a new approach towards load balancing in cloud-based content delivery networks. By applying adaptive server activations/deactivations at each data center in the cloud, overloaded data centers can move the extra load to lightly loaded ones without affecting the performance of any of the data centers or violating service level agreements (SLA) of users.. In addition to load balancing this method allows better resource management by adapting the number of active resources inside the data center to the offered load.

1 Introduction

Content Delivery Networks (CDNs) were introduced to allow for highly available and quick content delivery by keeping most recent content at servers near to users who usually request this data [1]. This becomes possible by distributing the task of data delivery on multiple centers in order to offload origin servers by delivering data on their behalf.

Recently, many CDN providers started migrating their networks into the cloud as the cloud provides numerous advantages for both CDN users and providers. The cloud helps reducing transmission latency as data is stored closest to the user. Operating costs are also reduced where resources could be rent from the cloud provider on demand. Cost reduction will also affect the users as they will no longer need to install physical storage devices to be part of the CDN, and will only pay for the content usage and content transfer [2].

However, there are many challenges that face cloud based CDNs, such as load balancing and network latency. Load balancing need to be done in parallel with locality awareness [3] to force users' requests to be routed to the nearest data center. It is also required to balance the load on data centers while minimizing the operation cost and maximizing the overall performance. Latency should be reduced as possible by caching data at non-origin servers nearest to end users who request it the most.

2 Approach

This study introduces an approach to load balancing in cloud based CDNs by loading data centers up to a certain threshold where delays are not affected instead of equally loading all data centers. In previous work by the authors [4],[5] a new algorithm was introduced to adapt the number of active servers in any data center to the amount of offered load by application of a multi-level, parallel hysteresis threshold algorithm. Results have shown that by applying this algorithm the delay experienced by users observe an almost constant behavior over a wide range of offered load. This property will allow user requests to be always routed to the nearest data center storing the requested data until the load on the data center reaches a certain threshold, afterwards additional servers may be activated or requests are routed to the nearest under-loaded data center.

The model used by the load balancing method in this context is named "Multiple Parallel Hystereses Model" which adapts the number of active servers at any data center to the offered load automatically by allowing activations and deactivations of servers only at certain thresholds. Figure 1 shows a state transition diagram for the model where each state has two variables (x,z); x is the number of active servers and z is the number of buffered frames.

Stationary state probabilities $P(x,z)$ are solved by an efficient iterative recursion algorithm under Markovian traffic assumptions. Detailed steps, formulas as well as delay calculations can be found in [4]. The model and the recursive algorithm can be generalized to include the overhead with server activations.

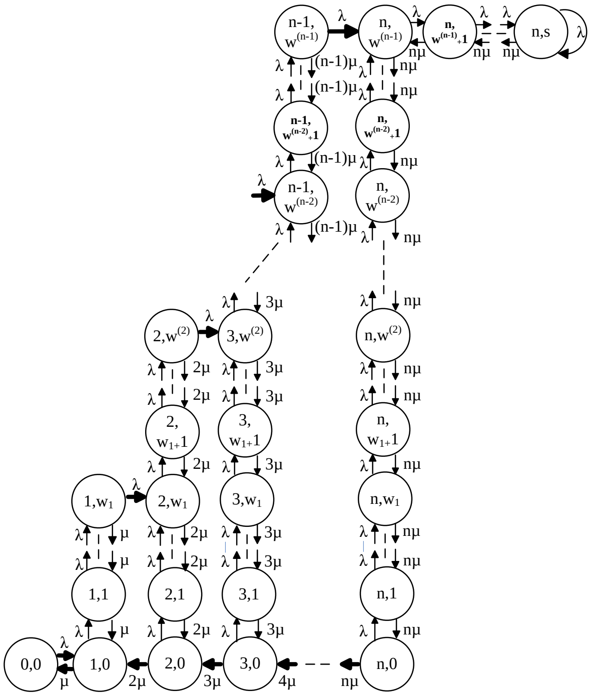

Fig. 1. State Transition Diagram for the Multiple Parallel Hystereses Model. Bold-faced transitions indicate server activations/deactivations

The proposed load adaptive algorithm allows for increasing the load on the data center up to utilization factor of 95%, so requests can always be routed to the nearest data center without lowering the quality of offered service.

The following assumptions are made for the load balancing process: N data centers are involved in the load balancing process, each has n_i servers and offered load = A_i where i=1,2,…..,N

The algorithm steps are as follows:

1. Determine the maximum load that could be handled by each data center

   ρ_{max}=[function (n) | $t_w < t_{SLA}$]

   where ρ_{max} is a function of the number of servers in each data center n_i and the maximum tolerable delay according to the users' SLA (t_{SLA}).
2. Determine the load margin

   $\Delta A(i) = A_i - \rho_{max}$

   If $\Delta A(i) > 0$: Data center i is overloaded and the extra load $\Delta A(i)$ needs to be shifted.

   If $\Delta A(i) \leq 0$: Data center i can still handle extra load equal to $\Delta A(i)$.
3. For DCs whose $\Delta A(i)$ >0, shift this amount of load to the nearest DC which can accommodate this load shift, fully or partially.
4. Repeat the above three steps until no more load shifting is necessary.

This algorithm could be applied centrally by sending $\Delta A(i)$ of each DC to a center station which takes decisions on where to shift any extra load. Alternatively, it could also be applied in a decentralized manner where data centers broadcast their $\Delta A(i)$ so that overloaded data centers could make decisions on where to shift the extra load if all data centers apply the identical algorithm.

4 Evaluation

Provided below is an example case for the multiple parallel hystereses model applied on a data center having 100 servers and 200 buffer places. Results are shown for hysteresis width w =1 and 2 for all hystereses. Figure 2 shows the average waiting time of buffered frames versus offered load. The average delay is almost constant over a wide range of values, which allows increasing the offered load without affecting the average delay.

The average delay values rise with the increasing the hysteresis width w, as the increase in hysteresis width implies buffering more frames before triggering activation of a server, more results on this issue are shown in recent publication by the authors [4].

Applying the proposed load balancing algorithm on several data centers in the cloud will lead to routing some users' requests to a data center that is not the nearest if the nearest one is highly loaded. This will increase the transfer delay because data will travel a longer distance. However, these delays are much smaller than the delays that the user could have experienced if requests were still routed to the highly-loaded data center.

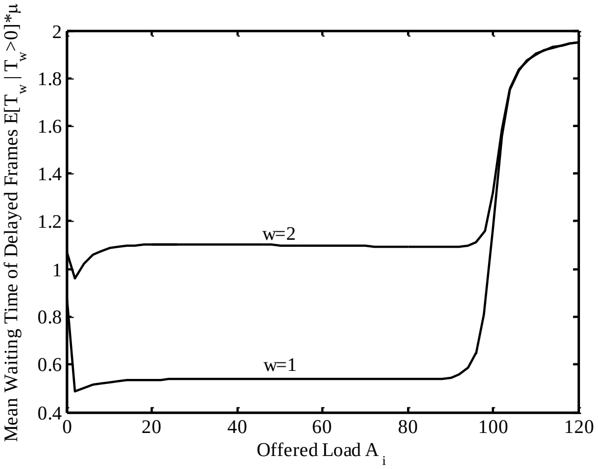

Fig. 2. Mean delay time for delayed frames versus offered load

5 Conclusion

This study introduced an approach to load balancing in cloud based content delivery networks. The approach takes advantage of the delay plateau provided by applying the multiple parallel hystereses model on each data center, which allows users' requests to be routed to the nearest data center while guaranteeing limited delays even if the data center is loaded up to its near capacity limit. Also shifting of any extra load from overloaded servers to underloaded ones could be done without affecting performance or users' service level agreements in terms of delay.

For future research this approach will be further extended to include deactivation of whole data centers for the process of load balancing in order to allow for balancing load as well as power reduction. The method allows further to extend the data center model to more realistic data center architecture consisting of multiple processing and storage servers and storage area networks (SAN).

References

1. Lazar, I., Terrill, W.: "Exploring Content Delivery Network," IEEE IT Professional, vol. 3, no. 4, 2001, pp. 47–49.
2. Wang, Y., Wen, X., Sun, Y., Zhao, Z., Yang, "The Content Delivery Network System Based on Cloud Storage," Network Computing and Information Security (NCIS), 2011 International Conference on , vol.1, no., pp.98-102, 14-15 May 2011
3. Lin, C., Leu, M., Chang, C., Yuan, S.: "The Study and Methods for Cloud Based CDN," Cyber-Enabled Distributed Computing and Knowledge Discovery (CyberC), 2011 International Conference on , vol., no., pp.469-475, 10-12 Oct. 2011
4. Kuehn, P.J., Mashaly, M.: "Modeling and Performance Evaluation of Self-Adapting Algorithms for the Optimization of Power-Saving Operation Modes", Proc. 1st European Teletraffic Seminar (ETS), Poznan, Poland, February 14-16, 2011
5. Kuehn, P.J.: "Systematic Classification of Self-Adapting Algorithms for Power-Saving Operation Modes of ICT Systems", 2nd Int. Conf. on Energy-Efficient Computing and Networking (e-Energy 2011), New York, USA, May 30 - June 1, 2011, ACM Digital Library