USING ADAPTIVE SERVER ACTIVATION/DEACTIVATION FOR LOAD BALANCING IN CLOUD-BASED CONTENT DELIVERY NETWORKS

Darshna Dalvadi[1] and Dr. Keyur Shah[2]

[1]School of Computer Studies, Ahmedabad University, Ahmedabad , India.
darshnadalvadi@gmail.com
[2] Head of the Department, L.D.R.P Institute of Technology and Research, Kadi Sarv Vidhyalaya, Gandhinagar, India.
profkeyur@gmail.com

ABSTRACT

Content Delivery Networks have been widely used for many years to serve million of users. Lately, many of these networks are migrating to the cloud for its numerous advantages such as lower costs, availability and dynamism in resource provisioning for obtaining overall increased performance. This paper introduces a new approach towards load balancing as well as lower response time(limited latency) in cloud-based content delivery networks allowing for providing overall improved performance within the tradeoff between power consumption and quality of experience (QoE). New proposed adaptive server activation/ deactivation model aims towards switching off unutilized servers at the data center to reduce the power consumption and also to use available resources at maximum efficiency while maintaining its performance. Latency can be lowered by only shifting the load off a datacenter when it is almost fully loaded.

KEYWORDS

Cloud Computing, Content Delivery, Load Balancing, Power reduction, Resource Management, Adaptive Self Control, Performance Modeling.

1. INTRODUCTION

This CDN is aimed towards serving end-users a large fraction of the Internet content like text, graphics and scripts, media files, software, documents, e-commerce, portals, live streaming media, on-demand streaming media, and social networks with good QoE. CDNs allow for limited latency by distributing content at servers across different region or worldwide and serving users by considering their geographical location, Origin of web page and type of requested data. Many distributed servers are doing jobs on behalf of original content server. Content is served by nearest sever with maintaining load balancing. Recently, many CDN providers started integrating their networks with the cloud as the cloud provides numerous advantages for both CDN users and providers. Using cloud CDN can be easily extended as resources could be rent from the cloud provider on demand. Also users will be benefited 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 which reduces the operating costs rapidly. However , there are two issues needed to be considered in cloud based CDN: 1) load balancing 2) network latency. Load balancing among

Dhinaharan Nagamalai et al. (Eds) : CSTY, SIGI, AI, MaVaS - 2015
pp. 21–25, 2015. © CS & IT-CSCP 2015 DOI : 10.5121/csit.2015.50903

different data centers in the cloud need to be done in parallel with locality awareness. This provides least delay in transferring content. A good performance metric should be considered while communication is latency which can be minimized by caching contents at different servers instead of serving from origin server. This study introduces an approach to load balancing in cloud based CDNs by offloading data centers when it is loaded beyond certain threshold and also maintaining delay bounds instead of equally distributing workload among servers so that data can be stored on data centers nearest to users who usually request it. In previous work a new algorithm was introduced to adapt the number of active servers in any datacenter accordingly the amount of offered network load at any time by using multi-level, parallel hysteresis threshold algorithm[1][2]. Results have shown that by applying this algorithm to any multiple server data center, the delay experienced by users observe an almost constant behavior over a vast range of offered load. This implies that if the Load increases up to a certain limit at a data center, the latency of content delivery can be limited and adapted to the users' SLA accordingly. By using this property provided by the algorithm, user requests will always be routed to the nearest data center storing the requested data until the load on the data center reaches a certain threshold, afterward requests are routed to the nearest under-loaded data center.

2. Adaptive Server Activation/Deactivation Model

A. Model without considering Server Activation Overhead

A load balancing method used in this paper is called "Multiple Parallel Hysteresis Model". This method adjusts the number of active servers according to offered load by switching off and on only at a certain threshold. There are two different thresholds for activation and deactivation to take place. Consider M/M/n/s queue model. In this extended notation C/B/n/s; C=M indicates the arrival rate of requests as Markovian (i.e., negative-exponentially distributed inter arrivals times), B=M indicates negative-exponentially distributed service time*s, n denotes the total number of* servers, and denotes the finite number of buffers for requests (frames). The parameters λ and μ indicate the arrival and service rates, respectively. The load factor

$$A = \lambda/\mu \qquad (1)$$

indicates the average number of occupied servers as *A*<n. In a finite buffer system further request arriving is to be rejected when buffers are full. Each system state is represented by a pair(x,z)where *x* is the number of active servers and *z* is the number of buffered frames. Now, instead of activating server at each arrival of request, requests are buffered until certain threshold has reached, then only activation is done. Activation thresholds are determined by the number of buffered frames, namely $w^{(1)}$, $w^{(2)}$, $w^{(3)}$, …$w^{(n-1)}$, where $w^{(i)}$ is the width of the *i*th hysteresis and $w^{i} = w^{(i)} - w^{(i-1)} \geq 0$ indicates the increase in buffered units for x=i until the next server is activated, i=1,2,…,n-1 where $w^{(i)}=w^{1}+\ldots+w^{i}$ and $w^{(0)}=0$.These hysteresis widths can be adjusted in a such way to meet the SLA's within the power reduction/performance tradeoff scenario. Deactivation policy works similar to any M/M/n queue where deactivations take place only when a server becomes idle and no buffered requests remain. The stationary state probabilities *P*(*x*,*z*) which can be determined by selecting activation and deactivation threshold is discussed in [3]. Applying this model increases probability of selecting hysteresis state where amount of load is like X or X+1 servers are active, and reduces probability of all other states. Average delay experienced by delayed customers in this model is measured using Little's Theorem according to the following relations:

- Average number of delayed arrivals

$$L = \sum_{x=1}^{n} \sum_{z=1}^{s} P(x,z) \qquad (2)$$

- Probability of buffering an arrival

$$W = 1 - P(0,0) - P(n,s) \qquad (3)$$

where arrivals to the system at state (*0*,*0*) are served immediately and at state (*n*,*s*) are dropped.

- Mean waiting time of buffered arrivals

$$E[Tw|Tw > 0] = \frac{L}{\lambda W} \qquad (4)$$

- Probability of lost arrivals

$$B = P(n,s) \qquad (5)$$

B. Model with Server Activation Overhead

Practically, deactivated servers take some time to become activate depending on their current deactivation mode. If node is switched off (disabled power supply) so it has to be booted from starting, which takes more time. If server is set in a sleep mode then it takes lower time to be activated. If we consider this time in estimating average delay in above discussed scenario then calculation is as follows: when the sum of activate servers and number of buffered frames in all the states with lower number of busy servers, a server is triggered to become activate. The rate at which a server is triggered to be activated is α, then average overhead time for an idle server to become active is $^1/_\alpha$ add this delay factor in mean waiting time.

3. LOAD BALANCING STRATEGY

Implementation of the load balancing strategy explained in this paper shows that, the load adaptive model applied on each datacenter of CDN allows the data center to have almost fixed delays over a vast range of loads. This allows for increasing the load on the data center up to utilization factor of 95% and more, so requests are always routed to the nearest data center without lowering the quality of offered service even if the data center is highly loaded. Certain assumptions are as follows:

- Total N number of data centres are there
- Each data center has n_i servers
- Each data centre has offered load

$$A_i = \frac{\lambda_i}{\mu_i} \qquad (6)$$

The algorithm steps are as follows:

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

$$A_{(max,i)} = [function(n_i)|t_w < t_{SLA}] \qquad (7)$$

where $A_{(max,i)}$ is a function of the number of servers n_i in each data center *i* and the maximum tolerable delay according to the users' SLA t_{SLA}.

2. Determine the load margin

$$\Delta A(i) = A_i - A_{(max,i)} \qquad (8)$$

If $\Delta A(i) > 0$: Data center *i* is overloaded and the extra load $\Delta A(i)$ needs to be shifted to another data center. If $\Delta A(i) \leq 0$: Data center *i* can still handle extra load equal to $\Delta A(i)$ without affecting its performance.

3. For DCs whose $\Delta A(i) > 0$, shift this amount of load to the nearest DC who can accommodate this load shift, fully or partially.

4. Repeat the above three steps until no more load shifting is necessary.

The question is how to select the nearest datacenter which can handle the load shift fully or partially two approaches are there:

1. Centralized: Each datacenter sends calculated ΔA to the center entity(cloud manager) and this entity decides where to shift the load.

2. Decenterlized: Each datacenter broadcast their ΔA to other datacenters and overloaded datacenter decides where to shift the load. Here, as long as the load is within the stable delay region, all the datacenters works in a self-controlled manner and needs no longer external control mechanism. The activation-deactivation model is sufficient to handle stable delay within the performance and power reduction tread off.

4. CONCLUSION

Paper introduced a new model for load balancing in cloud based CDN. Model employs following approaches:

1. Apply multiple parallel hysteresis model on each datacenter for providing limited delay.

2. Model attempts to adapt number of active servers accordingly the offered load which causes delay to be constant over a vast range of load. This allows routing users' requests to the nearest data center while maintaining the maximum load it can handle.

3. It also does load balancing by shifting any extra load from overloaded datacenter to the nearest neighbors which can handle this load coming from other data center without affecting the SLA's of users.

The model forces the data center to adapt the number of active servers to the offered load which causes the delay value to be constant over a vast range of values. This allows routing users' requests to the nearest data center to guarantee limited delays even if the data center was highly loaded. Also shifting of any extra load from overloaded servers to other under-loaded ones could be done without affecting their performance or affecting the users' service level agreements. The proposed approach balances the load between different cloud-based data centers while maintaining low delays. The model parameters can be adapted such that the additional delay resulting from buffering still meets SLA requirements.

REFERENCES

[1] 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

[2] Kuehn, P.J.: "Systematic Classification of Self-Adapting Algorithms for Power-Saving Operation Modes of ICT Systems", submitted in contribution to the 2nd Int. Conf. on Energy-Efficient Computing and Networking (e-Energy 2011), New York, USA, May 30 - June 1, 2011

[3] Kuehn, P.J., Mashaly, M.: "Performance of Self-Adapting Power-saving Algorithms for ICT Systems", Forthcoming paper (submitted)

AUTHOR

Darshna Dalvadi received Master of Computer Applications from H.N.G University, in 2007. Working as a lecturer at Ahmedabad University and pursuing Ph.D. from CUSHAH University.

Dr. Keyur shah is working has a Head of the Department at L.D.R.P, Institute of Technology and Research, Gandhinagar. He is having 12 years of experience in academic field also guiding no. of Ph.D research scholars.