

# Modeling and Evaluating of User Access in Content Delivery Network

Long Zhan<sup>1, a</sup>, Leijie Sha<sup>2, b</sup>, Rui Wang<sup>3, c</sup>, and Yang Liu<sup>4, d</sup>

<sup>1,2,3,4</sup>School of Information Engineering, Yangzhou University, Yangzhou, China

<sup>a</sup>1099200295@qq.com, <sup>b</sup>leijie.sha@qq.com, <sup>c</sup>ruiwang@qq.com, <sup>d</sup>yznsyly@163.com

**Keywords:** CDN, PEPA, Performance Evaluation

**Abstract.** Content Delivery Network (CDN) is a new kind of Internet content service system, which is built based on IP network and can provide the content distribution and service to satisfy the requirement of content access and application efficiency, the requirement of quality and contents order. The basic idea of CDN is to avoid bottlenecks and link which may affect data transmission speed and stability on the Internet, to make the content transmission faster and more stable. In this paper, we use a highly formalized method, named Performance Evaluation Process Algebra (PEPA), to model, analyze and evaluate the working principle of the CDN. By modeling and analyzing, it can be obtained that CDN can reduce the response time of user access and prevent the network congestion. Simulation results help to prove our analytical results.

## Introduction

In the early development of the Internet, network service capacity requirements are not that high and a lot of websites can meet the needs of the server. With the growth of the network content and the number of users, these websites will soon face the bottleneck of the service. In order to solve these problems, content delivery service technology emerges in responding to the proper time and conditions. Content Distribution Network (CDN) [1] is a location based on server replica and mechanism request redirection, which guarantees the availability of resources, the quality of service and the proximity of content to the user and can plus an efficient and content based routing. CDN is based on the following principles: 1. Choose the best equipment to provide users with services; 2. If a content required by many users, the closest user cache node is chosen. Based on these principles, user response time will be greatly reduced, and can solve the problem that load ability of inter operator server is too low while crossing regional.

Stochastic Process Algebras has great advantages in modeling concurrent systems, especially there are multiple concurrent and parallel components in the systems, which can be got in [2]. Performance Evaluation Process Algebra (PEPA) is shown in [3], which is a simple stochastic process algebra with powerful modeling ability, proposed by Hillston in the 1990s. PEPA has been used successfully in many fields, which can be seen in [4][5][6], which draws my interest to motivate the research of modeling CDN with PEPA.

The rest of this paper is structured as following: Section Two shows a background including introduction for PEPA and CDN. Section Three shows the PEPA model of CDN. Section Four presents the evaluation and analysis of the model. Section Five concludes this paper.

## Background

Performance Evaluation Process Algebra (PEPA), which is summarized by Hillston in the 1990s, has a strong ability to build a model, assessment and analysis of large-scale concurrent system. This part presents a brief introduction for PEPA. Compared with general process algebra, the action is assumed to have a duration, the time is subject to exponential distribution of random variables and the underlying theory of PEPA model is CTMC. Thus each action expressed in PEPA is a pair  $(\alpha, r)$ , where  $\alpha$  means the type of action, and  $r$  means the rate of activity. The following is a brief introduction for PEPA.

1) Syntax: the structure operation semantics can be found in [9]. The grammar is as follow:

$$S ::= (\alpha, r).S \mid S + S \mid C_s$$

$$P ::= P \underset{L}{\bowtie} P \mid P/L \mid S \mid C_p$$

As we can observe, there are two type of components in PEPA:  $S$  means a consecutive component and  $P$  means a model component that implements in parallel.  $C$  is constant and the constants for the two components are  $C_s$  and  $C_p$  respectively. The difference of  $C_s$  and  $C_p$  is that the cooperation just can be allowed between sequential component.

2) Semantics: Prefix: the prefix component  $(\alpha, r).S$  means the behaves as  $P$  after carrying out the activities  $(\alpha, r)$ , which contains the type of action  $\alpha$  and a duration that meets exponential distribution with parameter  $r$ .

Choice: The component  $P + S$  means a system may behave either as  $P$  or  $S$ . The actions of both  $P$  and  $S$  are enabled. Every of them has a related rate. There are race conditions for them, and they select the first to complete. If the activity belongs to  $S$ , the system may behave as the derivative of  $S$ ; and vice-versa for  $P$ .

Cooperation:  $P \bowtie_L S$ : the execution two processes  $S$  and  $P$  are concurrent and  $L$  is a collection of actions. The  $L$  needs  $S$  and  $P$  to complete the same time. e.g., if  $L = \emptyset$ , it notes that  $P$  and  $S$  do not need to synchronize any actions, expressed as  $P || S$ .

Hiding:  $P/L$ : All the actions in the collection  $L$  are invisible to external observers, and the system is in the process of  $P$  when all types of  $L$  are hidden.

Constant:  $A \stackrel{def}{=} P$ : The process  $P$  is assigned to the constant  $A$ , which means the constant  $A$  performs the similar behavior of the process  $P$ .

## CDN AND MODLLING

### A. Content Delivery Network

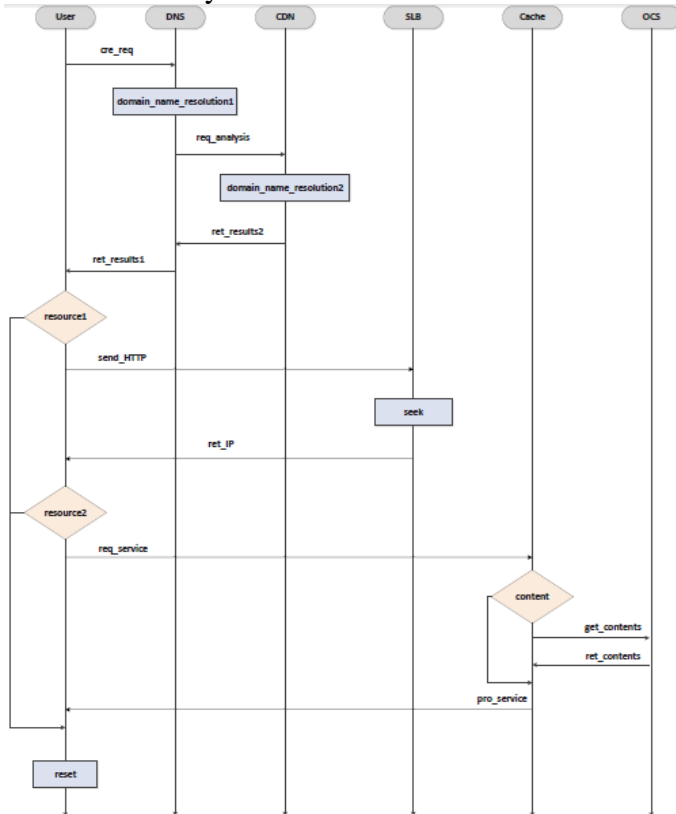


Fig. 1. The process of Content Delivery Network

Figure 1 presents the process of CDN. The process of user has access to web sites in the case of CDN begins with user making the request of resolving domain name to local DNS ( $cre\_req$ ). The local DNS and the authoritative DNS parse the domain name ( $domain\_name\_resolution1$ ). After that, the authoritative DNS return the CNAME for domain name to the Local DNS. Local DNS makes the

request of parsing domain name to the CDN DNS server(*req\_analysis*). CDN DNS server parses the domain name(*domain\_name\_resolution2*), CDN DNS server returns the results of the domain name resolution to local DNS (*ret\_results1*), and Local DNS returns this to user (*ret\_results2*). According to the results of domain name, CDN determines whether there exists resource constraints(Customer legitimacy,Acceleration mode,etc...). According to the IP address of local DNS, CDN determines the nearest user, and the IP address of the optimal SLB (Server Load Balance) is selected after the comprehensive judgment (*resource1*). If there exists resource constraints, user will create the request of domain name resolution again. If there is no resource constraints, user can make http content request to SLB (*send\_http*). SLB seeks the state of every Cache server in this area (*seek*), and SLB returns the IP address of the Cache server to user (*ret\_IP*). According to the domain name , SLB determines whether every Cache server in this area exists resource constraints (*resource2*). After the comprehensive consideration, if there exists resource constraints, then the user will create the request of domain name resolution again. Otherwise, SLB presents the optimal Cache IP address of the POP node, and returns the HTTP message to the user which includes a new CACHE IP address. The user requests the Cache server which is identified by the CACHE IP address to provide services (*req\_service*), Cache server to complete the request URL to store the contents of the map and determine whether the content in the local hit(*content*). If hits, Cache provides services directly to the user(*pro\_service*).If not, Cache gets related contents from OCS (*get\_contents*). Through the above process, the process of user access to web sites of CDN is completed.

#### B. PEPA Modeling of Content Delivery Network

This subsection presents a PEPA model of Content Delivery Network, which is composed of six main components. In order to make better analysis, we separate two parts from the CDN,i.e., the Cache server and Server Load Balance.

User: User performs first action *cre\_req* when User make the request of local DNS. The shared action *ret\_results1* is the local DNS that returns the results of the domain name resolution. The action *resource1* is a choice to verify whether there exists resource constraints. If there exists constraints, user will return to its initial state. Otherwise, the action *send\_http* means that user can make request for http content to SLB. After that, SLB returns the IP address of the Cache server by action *ret\_IP*. The action *resource2* is also a choice to judge whether every Cache server in this area exists resource constraints. If there exists, user will also return to its initial state. On the contrary, the action *req\_service* shows that user makes the request for server to the IP of CACHE identified Cache server. Cache server receives service and provide services to user directly by action *pro\_service* after getting the related contents.

$$\begin{aligned}
User_{cre\_req} &\stackrel{def}{=} (cre\_req, r_{cre\_req}).User_{ret\_results1} \\
User_{ret\_results1} &\stackrel{def}{=} (ret\_results1, r_{ret\_dom\_name}).User_{reset} \\
User_{resource1} &\stackrel{def}{=} (resource1_1, r_{resource1_1}).User_{send\_http} + (resource1_2, r_{resource1_2}).User_{reset} \\
User_{send\_http} &\stackrel{def}{=} (send\_http, r_{send\_http}).User_{seek} \\
User_{seek} &\stackrel{def}{=} (seek, r_{seek}).User_{ret\_IP} \\
User_{ret\_IP} &\stackrel{def}{=} (ret\_IP, r_{ret\_IP}).User_{resource2} \\
User_{resource2} &\stackrel{def}{=} (resource2_1, r_{resource2_1}).User_{req\_service} + (resource2_2, r_{resource2_2}).User_{reset} \\
User_{req\_service} &\stackrel{def}{=} (req\_service, r_{req\_service}).User_{ret\_contents} \\
User_{pro\_service} &\stackrel{def}{=} (pro\_service, r_{pro\_service}).User_{reset} \\
User_{reset} &\stackrel{def}{=} (reset, r_{reset}).User_{cre\_req}
\end{aligned}$$

Using the PEPA semantic definition of other components are similar, and thus bypassed. The system equation shows the construction of the defined components by forcing the cooperation

between them is on some action type. In order to simplify the formula, the parallel composition of M component is expressed as

$$C[M] := \underbrace{(C || \dots || C)}$$

The system equation can be written as

$$\text{User}[a] \bowtie_{S_1} (\text{DNS}[b] \bowtie_{S_2} \text{CDN}[c] || \text{SLB}[d] || \text{Cache}[e] \bowtie_{S_3} \text{OCS}[f]),$$

Where

$$S_1 = \{\text{cre\_req}, \text{ret\_result}, \text{send\_http}, \text{ret\_Cache\_IP}, \text{req\_service}, \text{pro\_service}\},$$

$$S_2 = \{\text{req\_analysis2}, \text{ret\_results2}\},$$

$$S_3 = \{\text{get\_contents}, \text{ret\_contents}\}.$$

### C. Parameter Settings

Table 1: Rate of actions (unit of duration: seconds)

Action	Description	Duration	Rate
cre_req	User submits the request of parsing	0.02	50
domain_name_resolution1	The local DNS and the authoritative DNS parse domain name	0.05	20
req_analysis	The local DNS makes the request to CDN DNS server	0.02	50
domain_name_resolution2	The CDN DNS server parses domain name	0.05	20
ret_results1	The local DNS returns the results to user	0.01	100
ret_results2	The CDN DNS server returns the results to the local DNS	0.01	100
resource1_1	CDN judges that there exists no resource constraints	0.001	1000
resource1_2	CDN judges that there exists resource constraints	0.001	1000
send_http	User sends http content request to SLB	0.05	20
seek	SLB seeks the state of every Cache server in this area	0.08	12.5
ret_IP	SLB returns the IP address of the Cache server to user	0.02	50
resource2_1	SLB judges that every Cache server in this area there exists resource constraints	0.001	1000
resource2_2	SLB judges that every Cache server in this area there is no resource constraints	0.001	1000
req_service	User makes the request of service to the Cache server	0.05	20
content1	Content in local hits	0.001	600
content2	Content in local no hits	0.001	1400
get_contents	Cache server gets related contents from OCS	0.02	50
ret_contents	OCS returns related contents to Cache server	0.5	2
pro_service	Cache server provides services to the user	2	0.5
reset	User waits for a period of time to create the next request	59	0.01
		7	

Table 2: Number of components

Component	Number
User	450
DNS	100
CDN	100
SLB	100
Cache	5
OCS	3

Table 1 and Table 2 present all the parameters about the performance evaluation, which includes the rates of actions and the number of different components. Most of rates in Table 1 are referred to the book named Dissecting Content Delivery Network. However, a small part of the rates is associated with the user action which can not be achieved by experiments. In order to make our analysis fruitfully, we set up a large number of components in Table 2.

### PERFORMANCE ANALYSIS

Response time is the main performance index of this system. To get the result of the analysis, we obtain the stochastic simulation of the CTMC by Matlab, which is derived from the PEPA. The following subsection will show more depictions about those fields which are available in[7].

#### Response time analysis

Response time means the duration from the first action asks for service to the last action receives the reply. In this subsection, we will consider the response time between *req\_service* and *pro\_service* in PEPA model.

Figure 2 presents the comparison of the response time of service's creation between different number of users. With the increasing of the number of users, the response time becomes longer. This situation may be caused by a main factor: the number of components in the CDN is certain. Such as DNS, SLB and OCS, increasing the number of users will lead to the congestion of Internet network,

and the reduction of load capacity of the server, which result into lower user access speed. Figure 3 presents increasing the number of OCS leads to a smaller response time. Figure 4 shows that when the number of Cache and OCS are in accordance with the proportion of 2 to 1 under the premise of the corresponding increase, and we find that the response time becomes smaller. Hence, it is necessary to increase the number of Cache and OCS to reduce the response time. Figure 5 illustrates that when the number of DNS and SLB increase, the response time is constant. So, 100 DNS serves and SLBs can meet the current state.

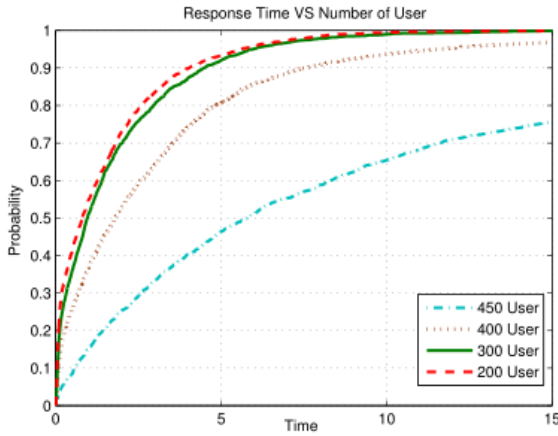


Fig. 2. Response time vs the number of User

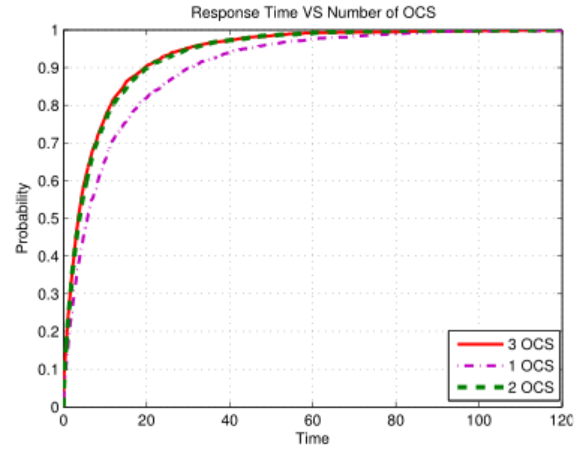


Fig. 3. Response time vs the number of OCS

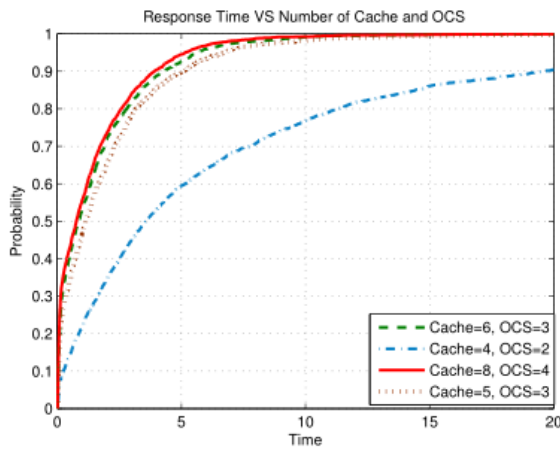


Fig. 4. Response time vs the number of Cache and OCS

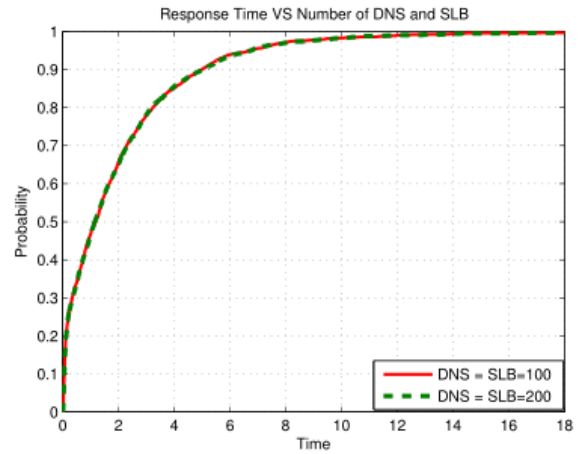


Fig. 5. Response time vs the number of DNS and SLB

## CONCLUSIONS AND FUTURE WORK

This paper has shown the PEPA modeling for the content distribution network, which uses the stochastic simulation of CTMC derived from the PEPA model to analyse the response time of different users. However, only the analysis of the response time of the users is considered and lack of the analysis of the throughput of Cache, so the model exists limitations. What's more, all parameters in the model are referred to the book named Dissecting Content Delivery Network. When the scale of the system is very large, the stochastic simulation can not be used to obtain the performances of the system. A better way is to overcome it is to use fluid approximation in[8][10] which can get acceptable performance easily without big loss of accuracy. In the future work, we will continue to improve the modeling of CDN and adopt the fluid approximation to analyze the large scale Content Delivery Network.

## Acknowledgments

The authors acknowledge the financial support by the National NSF of China under Grant (NO.61472343), and the NSF of Jiangsu Province of China under Grant (BK20140492).

## References

- [1] D. C. Verma: Content Distribution Networks, John Wiley, Sons Inc, (2002).
- [2] S. Donatelli, M. Ribaud, J. Hillston: A comparison of performance evaluation process algebra and generalized stochastic Petri Nets Proceeding of the Sixth International Workshop on Petri Nets and Performance Models, IEEE,(1995), pp.158-168
- [3] J.Hillston: A Compositional Approach to Performance Modelling, Cambridge University Press, (1996).
- [4] T.Tsang: Performance modeling and evaluation of millimeter-wave based WPANs, The 14th International Conference on Advanced Communication Technology (ICACT), IEEE, pp.142-147
- [5] J. Ding, J. Hillston, D. Laurenson: Performance Modelling of Content Adaptation for Personal Distributed Environment, Wireless Personal Communications, vol.48, (2008), pp.93-112
- [6] Y.Zhao, N.Thomas: Approximate Solution of a PEPA Model of a Key Distribution Center, Springer Berlin Heidelberg, (2008), pp.44-57
- [7] J. Ding, J. Hillston: Numerically Representing Stochastic Process Algebra Models, The computer journal, Vol 55 No 11,(2012), pp.1383-1397
- [8] J. Hillston: Fluid Flow Approximation of PEPA Models, Processing of the second International Conference on the Quantitative Evaluation of System, IEEE Computer Society, 2005, pp.33-43
- [9] R. A. Hayden: Scalable Performance Analysis of Massively Parallel Stochastic Systems, Imperial College press, (2011).
- [10] J. Ding: Structural and Fluid Analysis for Large Scale PEPA Models With Application to Content Adaptation System, Edinburgh University Press, (2009).