

ENHANCED EFFICIENT INFORMATION SHARING IN CLOUD ENVIRONMENT USING EERA

S.Sajithabanu,
Research Scholar,
Department of Computer Applications,
NIT, Trichy.
sajithabanu.phd.2013@gmail.com

Dr.S.R.Balasundaram,
Associate Professor,
Department of Computer Applications,
NIT, Trichy.
blsundar@nitt.edu

Abstract - Nowadays, Information storage and access through cloud network is becoming an economical alternative for delivery of traditional content. Mostly the content delivery is based on the moderate-size content providers. In Hybrid cloud network the services are be a subjected to momentous popularity increase due to its strategies and its performance of resources access. For optimization problem in cloud storage we have designed an EERA (Enhanced Efficient Replication Algorithm) for our optimization problem. The proposed work focused on reducing the bandwidth usage, traffic, delay access and latency also the performance cost and access time for storing and content delivery. Based on content provider the performance and results evaluation is take place in cloud environment. We implemented an effective and robust approximation algorithm in the proposed work and evaluated the complexity of the algorithm. The performance of the work provides a light-weight process, less complexity for time and space as well as cost, better accuracy of access of data and finally efficient distributed access in hybrid cloud environment.

Keywords: *Cloud storage, Content delivery, Replication.*

1 INTRODUCTION

Content Delivery Networks (CDNs) have progressed to overwhelm the intrinsic limitations of the Internet in terms of user alleged Quality of Service (QoS) when retrieving web content. A Content Delivery Network replicates video content from the origin server to proxy servers, disseminated over the world, so as to distribute content to end-users in a trustworthy and well-timed manner from nearby optimal proxies. It integrates development of high-end computing technologies with high-performance networking structure and dispersed replica management techniques.

Content Delivery Networks (CDNs) [1][2] provide services that improve network performance by increasing bandwidth, improving availability and retaining correctness through content replication.

They provide reckless and reliable applications and services by dispensing content to cache or edge servers located close to users [1]. A CDN has some mixture of content-distribution, request-routing, scattering and accounting infrastructure. The

content-distribution structure consists of a group of edge servers (also called surrogates) that distribute replicas of content to end-users.

2 COST OPTIMIZATION

As a traditional content distribution network consists of a central origin server and multiple proxy servers, a content distribution network over cloud storage is comprised of an origin server and multiple proxy servers on a cloud network. The proxy servers are connected with the origin server and/or other proxy servers, as illustrated in below figure. In order to shorten the latency experienced by the final content users, some objects are replicated on a proxy server, also called a replica, in advance near the users. Moreover, in order to improve the content users' experience, proxy servers may cooperate with each other so that they add objects from others rather than from the origin server. The proxy servers that are connected by a

direct overlay link can communicate with each

3 THE PUSH-PULL MODEL

Content users in area i will directly request and download object from the proxy server i . For simplicity of our discussion, we denote a proxy as if there is only one proxy server per one area. In practice, an origin server may be a server farm, and more than one proxy server may be in charge of one area. In such a case, the sum of proxy servers' storage space for area i can be used as c_i . Notice that cloud storage requires multiple replications based on predefined replication factor. The available storage space will be the actual storage size divided by the replication factor. Here, we assume that the same replication factor will be applied for all the contents, and use c_i as the available storage space for distinct contents. We point out that we focus on the performance with in a distribution cloud network and do not compute the latency and traffic costs between a proxy server and the final content users. In fact, those costs will not be impacted by the content distribution strategies, since each content user is directly served by a fixed proxy server according to the users' location. Therefore, our problem is to minimize the total cost, which includes the latency cost and the traffic cost between cloud servers, to satisfy all the requests from content users. Satisfying content users' requests, two typical schemes can be considered for content distribution, a push-based scheme and a pull-based scheme. In a push-based scheme, objects are replicated into proxy servers prior to requests until a proxy server's storage limitation is reached. This distribution procedure is referred to as a push.

4 CONTENT DELIVERY MODEL

When requests arise for those pushed objects, a proxy server will directly serve the content users without involving other cloud servers. On the other hand, no object is replicated into the proxy servers in advance under a pull-based

other, called neighbours.

scheme. When proxy servers receive queries for the un-pushed objects, they will forward the query and download the requested objects from the nearest source that could be either the origin server or another cooperative proxy server. This procedure is specified as a pull. By disseminating objects in advance, a push-based scheme shortens the retrieval latency of object j ; while a pull-based scheme reduces traffic volume by eliminating object downloads that would never be requested. In order to optimize both latency cost and traffic cost, we consider a content delivery scheme that is the combination of a push-based scheme and a pull-based scheme. This push-pull scheme is expected to properly determine which objects should be pushed and which should be pulled, so that both the push-based scheme and pull-based scheme are used to their best advantages.

5 RELATED WORK

A hierarchical framework is proposed and evaluated toward an efficient and scalable solution of content distribution over a multiprovider networked cloud environment [11]. A clustering data placement strategy that can automatically allocate application data among data centres based on [12] hybrid architecture for cost-effective streaming media distribution. The architecture combines two complementary technologies: CDN and P2P [13]. A link-level measurement on network provider platforms as a basis for network-wide traffic engineering, including load balancing for current trends and prognosis. The main goal, namely to shorten end-to-end paths and delays and reducing the load in the backbone and on expensive global interconnection links and reduce energy consumption[15].Measures the performance of the current Akamai platform [16] and considers a key architectural question faced by both CDN designers and their prospective customers whether the co-location approach to CDN platforms adopted by Akamai, which efforts to deploy servers in several Internet locations, brings intrinsic performance reimbursements over a more consolidated data

centre approach pursued by other influential CDNs such as Limelight[16]. To file transfer applications, heuristic distributed algorithm may find application in streaming services [19]. A Scaling decentralized content-based publish/subscribe (CBPS) networks proposed for large-scale content distribution [20]. The authors proposed a flexible agent-based modelling and simulation framework for designing and evaluating CDN architectures. It can be used to evaluate already existing solutions as well as to design and analyse novel solutions [21]. The authors proposed a simple structural approach for predicting Internet path latencies that out performs black box

techniques. Latency predictions are based on an underlying path prediction model that can predict PoP-level paths with high accuracy [22]. The authors proposed a latency-sensitive content distribution mechanism, Space4time, for a real world system. They proposed a novel content distribution and request routing solution, Space4time (space for time). Based on blocking probability, Space4time effectively exploits the storage and network capacity for latency-sensitive applications [23].

Table 1: Related Works

S.NO	AUTHOR	JOURNAL	YEAR	WORK DONE	LIMITATIONS
1.	Mohamed Diallo & at el	Elsevier	2012	Developed the publish/subscribe communication paradigm for content-based information retrieval which is used for improving communication efficiency.	Here the duplicate dropping heuristic does not thoroughly solve the problem addressed
2.	Xinjie Guan & at el	Elsevier	2013	Developed an approximation algorithm named Traffic-Latency- Minimization (TLM) for push-pull optimization problem which is used for content delivery from cloud storage.	DTLM algorithm may not find the most proper replica location for every object
3.	Shijia. Yao, & at el	JTAIT	2014	Developed two offline replica placement algorithms for cloud-based storage which is used to solved the load imbalance problems.	Proposed algorithms may not produce better results
4.	J.Sun, S.Gao & at el	JNW	2011	Developed an efficient algorithm (CPM) for allocate replicas of files to minimize the total cost. It includes three parts: replication algorithm pre-process, constraint P-median model and algorithm of solving constraint P-median problems which were solved by iteration methods .This CPM algorithm performs better than random algorithm with less total cost.	Limited Topology
5.	Sharrukh Zaman & at el	IEEE	2012	Developed a distributed approximation algorithm. The main focus is to improve the efficiency of popular object replication within a distributed replication group.	Degradation of system performance, not provide individual system performance, & number of replicated copies.
6.	Yan Chen, & at el	Springer	2002	A dynamic web content distribution system (d-tree) built on top of a peer-	Limited local network topology knowledge.

				to-peer overlay network called tapestry has need developed. This is used to find nearby replicas for the client	
7.	Samee Ullah Khan & at el	IJCISE	2009	Developed Fast Replica Placement in Content Distribution Network. Thus it reduces the make-span in the data replication.	Tree bottleneck problem.
8.	S.Zaman &D.Grosu	IEEE	2011	Developed Distributed Algorithm for the Replica Placement Problem which is used to Minimizes Response Time, Cost & Increase Availability.	Not provide individual system performance, & number of copies
9.	J.Sun, & at el	IIEEC	2010	Developed Replica Placement Algorithm in Content Distribution Network which is used to minimize total Cost.	Can't adapt dynamic user request
10.	S.Zaman &D.Grosu.	IEEE	2009	Developed Distributed Algorithm for Web Content Replication which is used to increase Response Time & Cost, Availability & Performance.	Not provide individual system performance, & number of replicated copies
11.	S.Ayyasamy & S.Natarajan	IJCSE	2009	Developed Intelligent Replica placement algorithm which is used to reducing both the access latency and Network traffic and increase the system performance.	High availability, Redundancy, Low Availability, Bottleneck Problem,
12.	N.Laoutaris & at el.	IEEE	2006	Developed a Two-Step Local Search (TSLs (k)) algorithm which employees Bloom Filters to distribute object, selected for replication in demanded patterns. . It reduces bandwidth, access cost.	Does not consider storage capacity, too many rounds, additional protocols & mechanisms.
13.	X.Tang & at el.	IEEE	2004	Developed tree based algorithm for solving the web server replica placement problem. Reduce the response time and cost & increase availability.	Does not provide Individual performance measurement
14.	Alain Roy & at el.	IEEE	2010	Developed ERMS, an elastic replication management system for HDFS. ERMS provides an active/standby storage model for HDFS. Enhance the reliability and availability of data.	Does not detect and predict the real-time data types
15.	A.Rajalakshmi & at el.	IJRSET	2014	Developed Dynamic Data Replica Selection and Placement in Hybrid Cloud and improved the overall data access performance and bandwidth utilization.	Does not extend replica placement in geographical locations

6 PROPOSED SYSTEM

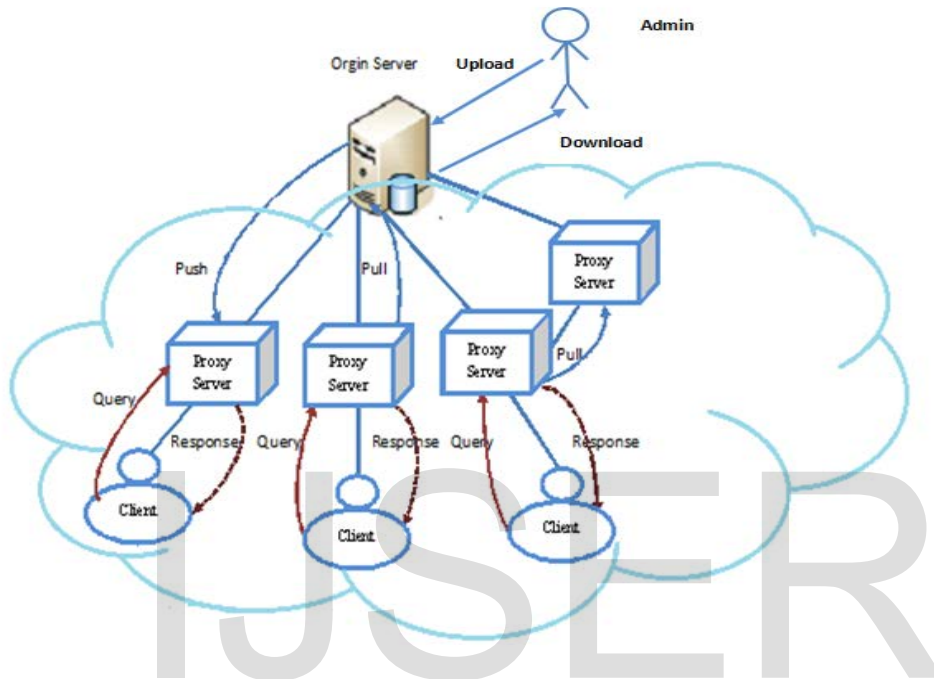


Fig. 1. Architecture Diagram Video Retrieval System

The users upload the data to the origin server. The same data would be replicated to the many proxy servers to increase the response time. The two mechanisms used here. The push and pull mechanisms used for the clients to retrieve the data fast and efficiently. The latency in area a to object b corresponds with the distance d_{ab} between proxy

server a and its nearest replica of b. The latency from a proxy server to different users may vary depending on a user's access network and the path. In this paper, we point on the latency from the location of a requested object to the proxy server that directly serves a final client, and omits the latency from this proxy server to the final client.

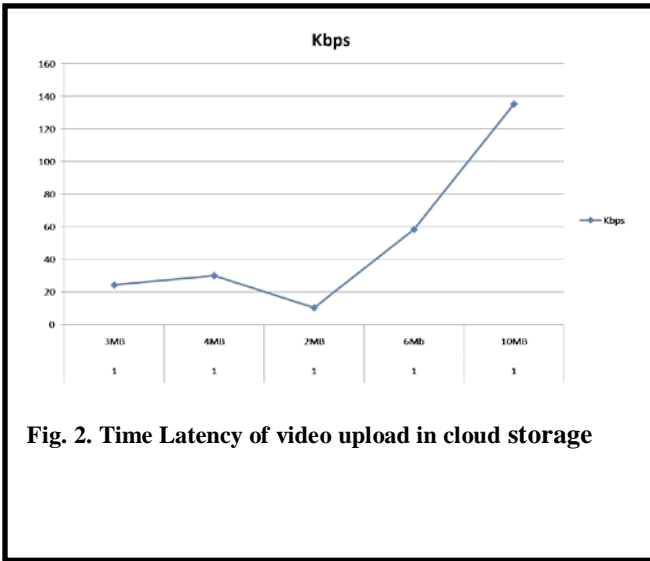


Fig. 2. Time Latency of video upload in cloud storage

Table 2:Notation

Notation	Explanation
p	The number of proxy server
q	The number of video objects
a	Index of proxy servers also indicates the service area for server a a=1,2,...p
c _a	The maximum storage space on proxy server a
b	Index of video objects, b =1,2,..q
d _{ab}	The distance from proxy server a to its nearest replica of video object b
λ _a	The total no.of request in area a.
P _{ab}	Probability that video object b will be queried in area a
S _b	The size of video object b
x _{ab}	Decision variable for pushing video object b to proxy server a
α	coefficient to balance the weight for startup time, latency & data traffic cost for pulling video object b in area a
k _{ab}	Total cost for a pull strategy with p proxy server and q video objects
Δ _{pq}	Δ _{pq} =∑ _{a=1} ^p ∑ _{b=1} ^q k _{ab}
f _{ab}	Cost saved by pushing video object b to proxy server a
F _{pq}	Total cost for push pull strategy with p proxy servers and q video objects including latency and data traffic cost

Our problem is to minimize the total cost that includes the storage cost, content update cost, latency cost and the traffic cost and bandwidth cost between cloud servers to satisfy all the requests from content users. The latency in area a to obtain video object b corresponds with the distance d_{ab} between the proxy server at its nearest replica of b. The latency from a proxy server to different users may vary depending on a user's access network and

the path. The start-up time and frame rate of streaming video calculated.

When a request is raised for video content object b in area a if b has already pushed in to the proxy server a , proxy server a can respond immediately without involving any extra latency. However if video object b has not been replicated on proxy server a in advance , proxy server a needs to pull video object b from either the origin server or another cooperative proxy server that results in a additional latency cost d_{ab} for each request for video object b in area a.

Assume that the total number of requests for video object b in area a is the product of λ_a , which is the total number of requests in area a and P_{ab} which is the probability that video object b will be queried in area a then the expectation of the total latency cost in area a is d_{ab}P_{ab}λ_a for pulled video object b.

$$X_{ab} = \begin{cases} 1 & \text{video object b is pushed on proxy server a} \\ 0 & \text{video object b is not pushed on proxy server a} \end{cases} \quad (1)$$

Now the expectation of latency in a cloud based CDNs using the push- pull mechanism can be represented as

$$\sum_{a=1}^p \sum_{b=1}^q (1 - X_{ab}) d_{ab} P_{ab} \lambda_a \quad (2)$$

And the total traffic volume for this cloud based CDN can be formulated as

$$\sum_{a=1}^p \sum_{b=1}^q (X_{ab} S_b + (1 - X_{ab}) S_b P_{ab} \lambda_a) \quad (3)$$

And the average bandwidth for this cloud based CDN can be formulated as

$$\sum_{a=1}^p \sum_{b=1}^q (S_b + Bf(t)) S_b P_{ab} \lambda_a \quad (4)$$

We used a coefficient α to balance the influence of latency and traffic costs and average bandwidth to satisfy the performance requirements.

Then our main objective is to minimize the total cost of latency and bandwidth and traffic and storage cost and content update cost for this network when using the Push-Pull mechanism under storage constraints.

$$\sum_{a=1}^p \sum_{b=1}^q \left((1 - X_{ab}) d_{ab} P_{ab} \lambda_a + \alpha \sum_{a=1}^p \sum_{b=1}^q (X_{ab} S_b + (1 - X_{ab}) S_b P_{ab} \lambda_a) \right) + \sum_{a=1}^p \sum_{b=1}^q (S_b + Bf(t)) S_b P_{ab} \lambda_a \quad (5)$$

Subject to storage constraints

$$\sum_{b=1}^q X_{ab} S_b \leq C_a \quad a=1,2,\dots,p \quad (6)$$

To solve this optimization process we simplify Eq.(5) and Obtain

$$\text{Min} \left(\sum_{a=1}^p \sum_{b=1}^q (d_{ab} P_{ab} \lambda_a + \alpha S_b P_{ab} \lambda_a - X_{ab} (d_{ab} P_{ab} \lambda_a + \alpha S_b P_{ab} \lambda_a - \alpha S_b)) \right)$$

Algorithm1. EER approximation algorithm – runs on origin server.

for every proxy server $a \in \{1, 2, \dots, p\}$
 Calculate f_{ab} and r_{ab} for each object $b \in \{1, 2, \dots, q\}$;
 Sort r_{ij} in descending order and record corresponding indices in array h ;
 Calculate total cost of system $\text{TotalCost} = \text{TotalCost} + \sum_{b=1}^q k_{ab}$;
 Keep a record of the weighted total cost for a simple pull system $\Delta_{pq} = \Delta_{pq} + \sum_{b=1}^q k_{ab}$;
 for index $i \in \{1, 2, \dots, q\}$;
 $b = h[\text{index}]$;
 if ($r_{ab} > 0$ AND $s_b \leq \text{available storage on } i$)
 Put b into PushSet_a ;
 $\text{TotalCost} = \text{TotalCost} - f_{ab}$;
 Compress the content and store in proxy server a ;
 else if ($r_{ab} > 0$ AND $s_b > \text{available storage on } a$)
 if ($\Delta_{pq} - f_{ab} < \text{TotalCost}$ AND $s_b \leq c_a$) Keep removing the last element in PushSet_a till there is enough room for b ;
 Update TotalCost ;
 Put b_j into PushSet_a ;
 end if
 else if $r_{ab} \leq 0$
 Break;
 end if
 end for
 end for

Algorithm2. DEER approximation algorithm – runs on each proxy server.

Initialize the distance d_{ab} for each object b as the distance from origin server to local;
 Initialize the location of each object b as origin server;
 Calculate the f_{ab} and r_{ab} for each object b ;
 Sort r_{ab} in descending order and record corresponding indices in array h ;
 Calculate the PushSet_i according to r_{ab} and current d_{ab} ;
 Update the location for object s in PushSet_a as local;
 Inform cooperative-proxy server about current PushSet_a ;
 While (receive PushSet_a ' from cooperative proxy server a')
 For each objects k in received cooperate-proxy server's PushSet_a '
 If distance from proxy server a' to local is nearer than d_{ab} ;
 Update d_{ab} as the distance from proxy server a' to local; record the location of object k as proxy server a' ;

Calculate the storage available storage space on as proxy server a' ;
 If storage space greater than received content size
 Receive the content from d_{ab} then compress the content and store in proxy server a' ;
 End if
 End if
 End for
 For each objects b
 If (Location is proxy server a') AND (Not in PushSet_a)
 Update the distance d_{ab} as the distance from origin server to local;
 Update the location of object b as origin server;
 End if
 End for
 Recalculate the f_{ab} and r_{ab} for each object b ;
 Recalculate the PushSet_a according to updated r_{ab} and current d_{ab} ;
 Inform cooperate-proxy server updated PushSet_a ;
 End while

7 CONCLUSIONS

We have analysed the problem of replica selection and placement that determines which contents should be pushed to which proxy servers and which contents should be pulled on demand for an optimal content delivery over cloud storage. Our work is first considering the both bandwidth usage in the network and latency for the optimization of content delivery using cloud storage services. We have replica and formulated this push-pull content delivery problem and have designed an EERA (Enhanced Efficient Replication Algorithm) for our optimization problem. The proposed work focused on reducing the bandwidth usage, traffic, storage cost, delay access and latency also the performance cost and access time for storing and content delivery and yield the better results.

8 REFERENCES

- [1] G. Pallis, and A. Vakali, "Insight and Perspectives for Content Delivery Networks," Communications of the ACM, Vol. 49, No. 1, ACM Press, NY, USA, pp. 101-106, January 2006.
- [2] A. Vakali, and G. Pallis, "Content Delivery Networks: Status and Trends," IEEE Internet Computing, IEEE Computer Society, pp. 68-74, November-December 2003

- [3] M. Day, B. Cain, G. Tomlinson, and P. Rzewski, "A Model for Content Internetworking (CDI)," Internet Engineering Task Force RFC 3466, February 2003. www.ietf.org/rfc/rfc3466.txt
- [4] M. Arlitt, and T. Jin, "A Workload Characterization Study of 1998 World Cup Web Site," *EEE Network*, pp. 30-37, May/June 2000.
- [5] J. Kangasharju, J Roberts, and K. W. Ross, "Object Replication Strategies in Content Distribution Networks," *Computer Communications*, Vol. 25, No. 4, pp. 367-383, March 2002.
- [6] S. Adler, "The SlashDot Effect: An Analysis of Three Internet Publications," *Linux Gazette Issue*, Vol. 38, 1999.
- [7] G. Pallis, and A. Vakali, "Insight and Perspectives for Content Delivery Networks," *Communications of the ACM*, Vol. 49, No. 1, ACM Press, NY, USA, pp. 101-106, January 2006.
- [8] F. Dougliis, and M. F. Kaashoek, "Scalable Internet Services," *IEEE Internet Computing*, Vol. 5, No. 4, 2001, pp. 36-37
- [9] Keynote Systems—Web and Mobile Service Performance Testing Corporation, <http://www.keynote.com/>, 2007.
- [10] The Gigaweb Corporation, <http://www.gigaweb.com/>, 2007.
- [11] Chrysa Papagianni, Aris Leivadreas, Symeon Papavassiliou , "A Cloud-Oriented Content Delivery Network Paradigm: Modeling and Assessment", *IEEE Transactions*, 2013
- [12] Dong Yuan, Yun Yang, Xiao Liu, Jinjun Chen," A data placement strategy in scientific cloud workflows", Elsevier, 2010
- [13] Dongyan Xu , Sunil Suresh Kulkarni, Catherine Rosenberg , Heung-Keung Chai ," Analysis of a CDN-P2P Hybrid Architecture for Cost-Effective Streaming Media Distribution", *IEEE* ,2010
- [14] Akamai Technologies Inc., "Akamai-The Business Internet - A Predictable Platform for Profitable E-Business," 2004.
- [15] Gerhard Haßlinger, Franz Hartleb," Content delivery and caching from a network provider's perspective", Elsevier , 2010.
- [16] Sipat Triukose Zihua Wen Michael Rabinovich," Measuring a Commercial Content Delivery Network ",ACM, 2011
- [17] B. Molina, C. E. Palau, and M. Esteve, "Modeling Content Delivery Networks and their Performance," *Computer Communications*, Vol. 27, Issue 15, pp. 1401-1411, September 2004.
- [18] S. Saroiu, K. P. Gummadi, R. J. Dunn, S. D. Gribble, and H. M. Levy, "An Analysis of Internet Content Delivery Systems," *ACM SIGOPS Operating Systems Review*, Vol. 36, pp. 315-328, 2002.
- [19] Yuh-Ming Chiu, IEEE, Do Young Eun, "On the Performance of Content Delivery under Competition in a Stochastic Unstructured Peer-to-Peer Network", *IEEE TRANSACTIONS*, 2010
- [20] Mohamed Diallo, Vasilis Sourlas, Paris Flegkas , Serge Fdida , Leandros Tassioulas ," A content-based publish/subscribe framework for large-scale content delivery ",Elsevier 2013.
- [21] Giancarlo Fortino, WilmaRusso,MarinoVaccaro," An agent-based approach for the design and analysis of content delivery networks", Elsevier 2011.
- [22] Harsha V. Madhyastha_ Thomas Anderson Arvind Krishnamurthy," A Structural Approach to Latency Prediction", *ACM* 2006.
- [23] Lingfang Zeng, Bharadwaj Veeravalli , QingsongWei," Space4time: Optimization latency-sensitive content service in cloud", Elsevier 2014.