KSII TRANSACTIONS ON INTERNET AND INFORMATION SYSTEMS VOL. 6, NO. 3, Mar 2012 954

CDN Scalability Improvement using a Moderate Peer-assisted Method

Peichang Shi[1]*, Huaimin Wang[1], Hao Yin[2], Bo Ding[1], Tianzuo Wang[1] and Miao Wang[3]
[1]National Laboratory for Parallel and Distributed Processing, School of Computer Science,
National University of Defense Technology
Chang Sha, 410073 – China
[e-mail: pcshi@nudt.edu.cn, whm_w@163.com, dingbo@nudt.edu.cn, phoenixwtz@163.com]
[2]Department of Computer Science and Technology, Tsinghua University
Beijing, 100084 – China
[h-yin@mail.tsinghua.edu.cn]
[3]Department of Computer Science and Engineering, University of Nebraska-Lincoln
Lincoln, NE 68588-0115 – USA
[mwang@cse.unl.edu]
*Corresponding author: Peichang Shi

Received October 19, 2011; revised February 19, 2012; accepted March 12, 2012; published March 25, 2012

Abstract

Content Delivery Networks (CDN) server loads that fluctuant necessitate CDN to improve its service scalability especially when the peak load exceeds its service capacity. The peer assisted scheme is widely used in improving CDN scalability. However, CDN operators do not want to lose profit by overusing it, which may lead to the CDN resource utilization reduced. Therefore, improving CDN scalability moderately and guarantying CDN resource utilization maximized is necessary. However, when and how to use the peer-assisted scheme to achieve such improvement remains a great challenge. In this paper, we propose a new method called Dynamic Moderate Peer-assisted Method (DMPM), which uses time series analysis to predict and decide when and how many server loads needs to offload. A novel peer-assisted mechanism based on the prediction designed, which can maximize the profit of the CDN operators without influencing scalability. Extensive evaluations based on an actual CDN load traces have shown the effectiveness of DMPM.

Keywords: CDN, peer-assisted, vertical scalability, prediction model

This research was supported by National Basic Research Programof China [No.2011CB302601, Basic Research on Effective and Trustworthy Internet-Based Virtual Computing Environment (iVCE)], by National NaturalScience Foundation of China [No.90818028, Behavior Monitoring and Trustworthiness-Oriented Evolution of Large-Scale Distributed Software Systems], by the National Science Fund for Distinguished Young Scholars [No.60625203, Computer Software].We express our thanks to Yizhi Song and TaoHu from the ChinaCache company in China specially, who have done a lot relevant and basic work to the topic. The authors alsothank the editor and the anonymous reviewers for their thoughtful comments.

DOI:10.3837/tiis.2012.03.011

1. Introduction

According to Computerworld News [1], the world's data will increase by 50 times in the next decade. Such large-scale data will delivered through Content Delivery Networks (CDN), mostly to geo-distributed end-users (e.g., Akamai, one of the CDN entities, routinely delivers between 15% and 30% of all Web traffic of the entire Internet). However, the service capacity of a CDN is limited. The under provisioning case [2] (i.e., when the end users' aggregated peak resource demand exceeds the CDN service capacity) occurs occasionally. As such, if a CDN does not have enough usable capacity to satisfy the exceeding resource demand, the exceeding part of end users' requests declined. Furthermore, some end-users give up on accessing the service permanently after experiencing poor Quality of Service (QoS) [1]; as a result, the CDN operators lose a portion of the profit permanently because of the loss of end-users. Therefore, improving CDN scalability is very important. The two important kinds of scalability are horizontal and vertical [3]. Horizontal scalability refers to the traditional load-balanced model, while vertical scalability refers to CDN service capacity improved through Peer-to-Peer (P2P) technology. Unless otherwise stated, the scalability referred to in the current paper refers to vertical scalability, which is the focus of our current research.

Many related studies conducted on the improvement of the vertical scalability of CDN. Xu et al. [4], Yin et al. [5], and Huang et al. [6] demonstrated the feasibility of improving the vertical scalability of CDN using the peer-assisted method. Karagiannis et al. [7] quantified the effect of the peer-assisted method on the service capacity of CDN to some degree. Pakkala et al. [8] presented CDN-P2P architecture, while Rodriguez et al. [9] delved into the feasibility of a commercial, legal P2P technology. These previous studies have mainly focused on the feasibility of commercially distributing peer-assisted content and improving CDN scalability for three kinds of applications, such as Video on Demand (VoD), live streaming, and file sharing, among others. Indeed, P2P is an effective method in improving CDN scalability. However, current methods only use peer-assisted method to offload the CDN server load regardless of CDN resource utilization and CDN operators' profit. Overusing P2P may result in the reduction of CDN resource utilization and the corresponding loss of a portion of operators' profit since resource utilization is proportional to the operators' profit. Previous studies could have been more reasonable if the authors considered this problem. Furthermore, none of the previous studies valued the end users' QoS and CDN resource utilization to find a win-win situation for both end-users and the operators. Thus, further studies are necessary to solve the problems regarding "when, how much, and how to use the peer-assisted method to improve the vertical scalability of CDN moderately".

The current paper proposes to improve CDN vertical scalability moderately as well as avoid the possible conflict between scalability improvement and resource utilization based on assured end-users' QoS. Therefore, we propose a new method called Dynamic Moderate Peer-assisted Method (DMPM), which predicts and decides when and how many server loads must cut down through the peer-assisted method. We then design a novel P2P mechanism to improve CDN scalability moderately. DMPM is a distributed method involving CDN dynamic load online prediction using time-series analysis method, a novel P2P mechanism using the View Upload Decoupling (VUD) method. DMPM solves the CDN vertical scalability problem, guarantees the end users' QoS, and assures maximization of CDN resource utilization. The following are the main contributions of this research:

- Present and formulate the problem regarding the improvement of CDN vertical scalability in a peer-assisted method from the perspective of CDN operators.
- Presentation of DMPM, which adopts the time series prediction model called

Autoregressive Integrated Moving Average Model(ARIMA)to predict when and how many CDN server loads need to be cut down, as well as design the proper P2P mechanism to improve CDN vertical scalability and guarantee end-users' QoS as well as CDN resource utilization.

- Use of traces collected from a real CDN entity, namely, China Cache [5], as benchmarks to evaluate DMPM and four performance metrics, including the Prediction Error (*PE*), CDN Resource Utilization (*CRU*), Request Response Rate (*RRR*) and Scalability Coefficient (*SC*) to prove its effectiveness.

The remainder of this paper organized as follows. Section 2 discusses the related works**,** Section 3 introduces the system model and problem formulation, and Section 4 proposes DMPM and presents the detailed algorithm of the CDN dynamic load's online prediction and popularity-aware P2P policy. Section 5 gives a comprehensive evaluation to demonstrate the effectiveness of this method, and Section 6 concludes the paper.

2. Related Works

CDN and P2P are two technologies used to deliver varied contents (e.g., Web, video, audio, etc.) to the end-users [4][5]. Both technologies have their own advantages and disadvantages. CDN provide excellent quality service to end-users when the workload is within its service capacity; however, CDN servers are expensive to deploy and maintain. In comparison, P2P achieves high scalability while maintaining the server requirements low; however, undesirable side effects may occur, such as QoS is unguaranteed and network is unfriendliness. In the current paper, two research areas related to our work: pure P2P mechanisms and the design of large-scale hybrid CDN-P2P [4].

Pure P2P Mechanisms. P2P has become a powerful technology in delivering contents to geo-distributed end-users for various applications, such as file sharing, VoD, and live streaming. Structured and unstructured P2P networks are the two main kinds of P2P networks used for file sharing. The former consists of a set of nodes interconnected in a regular topology, and the latter is interconnected arbitrarily [10]. Structured P2P networks built based on the distributed hash table abstraction, which guarantees the location of information in a small number of hops. However, maintaining the overlay network topology, such as that discussed in Chord [11], Pastry [12], and Kautz [13][14] require high cost. Conversely, unstructured P2P networks usually built based on a distributed and loose structure that is cost-efficient. However, if a research relies on a flooding mechanism, the localization of content guaranteed, as in the cases reported by Gnutella [15] and Freenet [16]. In pure P2P system, the P2P overlay can alleviate the server load effectively, as exemplified by a P2P on-demand streaming system [35], which uses a novel caching scheme called dynamic buffering. In most of the above-mentioned research, peers only share content that they can access, similar to VoD and live streaming P2P mechanisms in the literature [17][18][19].

Hybrid CDN-P2P. Hybrid CDN-P2P integrates the advantages of quality control and reliability in a CDN, and the scalability of a P2P system [36]. Xu et al. [4] first performed an in-depth analysis of the hybrid CDN-P2P architecture, for which they focused on the transition from the stage of CDN-P2P co-existence to P2P. Yin et al. [5] presented a multi-stage approach similar to [4] in the real-world hybrid CDN-P2P system called LiveSky. Karagiannis et al. [7] quantified the effect of peer-assisted file delivery on end-users experience and resource consumption, and then highlighted that the simplicity of "locality-awareness" of P2P delivery solutions could significantly alleviate the induced cost of the Internet Service Providers (ISPs). Meanwhile, Huang et al. [6] presented the potential savings in using hybrid

CDN-P2P systems in two major CDN entities, such as Akamai and Limelight, similar to Karagiannis et al. [7]. Rodriguez et al. [9] investigated the feasibility of commercial and legal P2P content distribution solutions. Pakkala et al. [8] presented aCDN-P2Parchitecture, which targets function enhancement and extends the reach of the legacy star topology-type of CDN using P2P techniques.

Most of the above-mentioned works are from the perspective of ISPs and content providers or consumers, and none of them views the problem from the perspective of CDN operators. Furthermore, most of the studies cited above have focused mainly on the following aspects: how to construct the P2P overlay, the key structure and protocol of P2P, the architecture of the hybrid CDN-P2P, the feasibility and effectiveness of using P2P to assist CDN, and the main phase and key mechanisms in CDN-P2P. In contrast, little attention has been devoted to improve scalability from the perspective of CDN operators. Given that improving the scalability of CDN using peer-assisted method is a very important issue in the era of cloud computing [37], thus complementing the current literature, our study focuses on the stage of CDN-P2P co-existence and the exact number of server loads need to be reduced by P2P technologies in the stage. The problem in the present study is not the same as that discussed in previous works, but the latter resolution may heavily depend on current research. For example, we adopted new ideas in P2P that enable peers to share any content they possess, including VUD mechanism [20], and various distribution mechanisms, such as trees [21], mesh-pull [22], and meshes with push-pull [23].

Other related studies are found in literature [33][34] in terms of predicting how much server load must be offloaded through P2P technology, which is important in improving CDN scalability. Niu et al. [33] used time-series analysis techniques to predict the online population automatically, the peer upload**,** and the server bandwidth demand in each video channel of peer-assisted on-demand video streaming. Isalm et al. [34] developed prediction-based resource measurement and provisioning strategies using neural network and linear regression to satisfy upcoming resource demand. The predicted objects of the two works are CPU utilization of e-commerce and bandwidth demand of peer-assisted on-demand video streaming respectively. Unlike the previous researches, our research object is the traffic behavior of CDN; however, whether or not this kind of resource demand behavior can predicted is uncertain. Furthermore, existing works and methods cannot use to achieve improving CDN scalability moderately.

3. System Model and Problem Formulation

3.1 System Model

We considered a CDN system with multiple geo-distributed CDN nodes, which are logical management units. Each CDN node comprises several physical servers in the same location and provides many different services to local end-users, such as Web applications, download applications, streaming, and dynamic applications (e.g., online games and social networks etc.). A CDN node is response for allocating resources within the network to different applications, server management, task allocation, resource scheduling, operation monitoring, and users' request processing, among others. **Fig. 1** illustrates the CDN architectures using a hybrid structure. Inside, a CDN node manages many servers that organized to form a star topology. Similar to that used by China Cache [28], the servers inside one CDN node are managed by one central server, and are usually divided into many groups, (with each group operating one kind of application). Furthermore, each kind of application (e.g. web application) comprises tens of customers (like different web sites) in China Cache. Thus, the servers inside the same group provide cache and deliver services to various customers. In order to enable different CDN nodes to obtain content from one another rather than only from the original server [5][8], all of the CDN nodes form a P2P topology through a private fiber [32]. For the current study, we focused mainly on improving the scalability of one CDN node using the distributed method, which shall discuss in the next section.

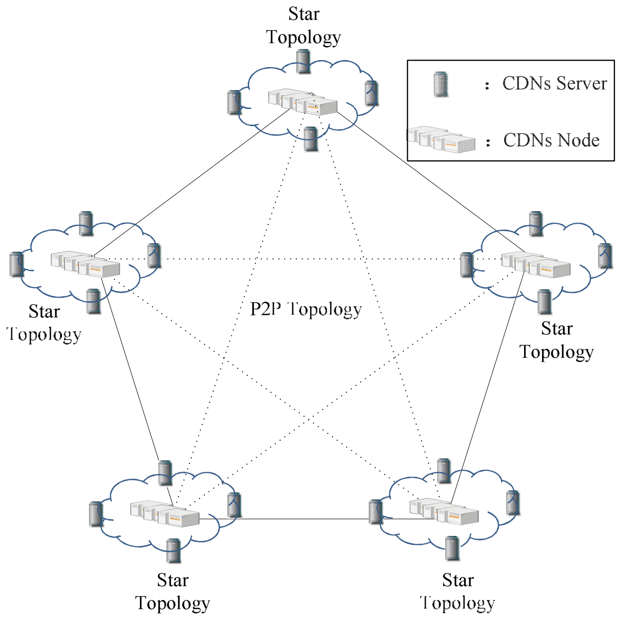


Fig. 1. CDN Architecture.

In a CDN, an end user's request routes to the optimal geo-distributed CDN node, depending on many factors, such as the network distance between the end-users and the CDN nodes, network congestion conditions and server loads on the CDN node, among others. When millions of users are directed to one local CDN node, and the aggregated resource demand exceeds the service capacity of the local CDN node, users can be organized to form P2P topology to diminish the local server load [4][5][6]. The users belonging to the same application or customer served by the same server inside the CDN node organized to form P2P overlay. Then the new requests for the same content provided by the P2P overlay can serve. From the perspective of CDN operators, they expect that the new request is served by the P2P overlay once the aggregated resource demand exceeds the service capacity of the local CDN node. The operators also expect the new requests to serve by local CDN node once the latter is resource utilization does not reach 100%. Other details about the CDN architecture and basic mechanism can be seen from previous works [4][32][36]. In the current work, we only want to state the related information about CDN scalability improvement.

Therefore, based on the CDN architecture and the main idea presented above, we focus on resource allocation inside a geo-distributed CDN node for multiple applications each possessing thousands of millions of end-users at the most. We considered the end-user assistance in content delivery when CDN needs to scale up. **Fig. 2** shows that in the period t_1–t_2, when end users' aggregated demand exceeds the service capacity of the CDN node, peer-assisted method is only adopted to offload the exceeding part inside the CDN node. This action improves CDN scalability, assures maximized CDN resource utilization, and guarantees end users' QoS. From the perspective of CDN operators, CDN scalability improved moderately. However, knowing when and how many server loads must cut down and the task of designing a corresponding effective P2P mechanism remain great challenges that need to address. In the following section, we present the problem formulation of CDN scalability improvement from the perspective of CDN operators.

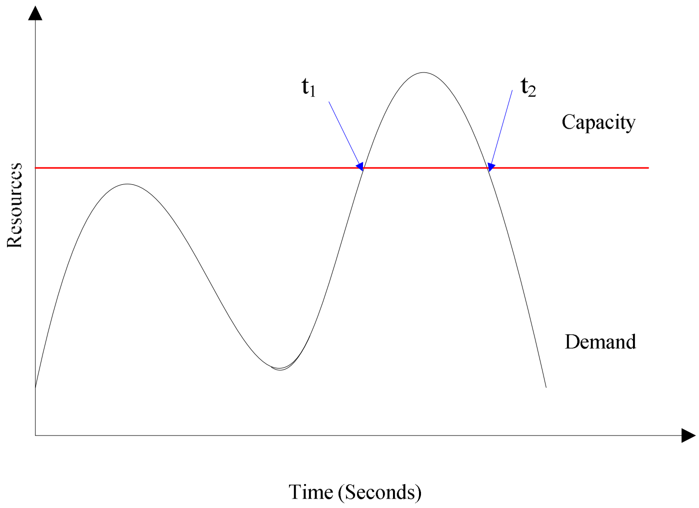


Fig. 2. Peer assisted method to improve CDN scalability.

3.2 Problem Formulation

This section describes the problem formulation for improving CDN scalability moderately in a peer-assisted, cost-efficient method. CDN is abstracted to a bidirectional graph *G*, which consists of a set of nodes *C* connected by a set of edges *E*. We selected one CDN node *c, c*∈*C*, as the research subject, because our method runs on a geo-distributed CDN node without any centralized control. The rated service capacity of each geo-distributed CDN node is *Rate,* and the node consists of a set of servers *S* and applications *A*. There are n_c^a end-users of application *a* in node *c*. The resource demand of each end-users is β_a on average, $\forall a \in A$, and $\forall c \in C$. Let s_c^a denote the resource allocated to application *a,* and q_c^a denote the service quality of application *a* in node *c*. Clearly, $\sum_{a \in A} S_c^a$ cannot exceed the limited service capacity *Rate* of node *c*. The higher the percentage of satisfied end users' requests for application *a* are the higher the q_c^a would be.

However, the numbers of end-users are unable to control or predict accurately by CDN operators. Thus, it is impossible for a CDN to provide continuously sufficient service capacity to end users, especially when the flash crowd phenomenon [2] occurs. At each time *t* when $\sum_{a \in A} n_c^a * \beta_a$ exceeds *Rate*, many end-users cannot obtain guaranteed quality service from the CDN node. Thus, in order to cut down the load of CDN servers under this situation as well as improve CDN scalability, we adopted P2P technology to organize part of the end-users (i.e. those who have already been served by CDN servers and have the ability to become seeds [4]) so that they can contribute in offloading the current CDN server loads. R_{P2P} denotes the service capacity formed from the end-users use of P2P technology when end-users' requests exceed the rated service capacity of the CDN. R_{P2P} equals $\sum_{a \in A} n_c^a * \beta_a - Rate$ when $\sum_{a \in A} n_c^a * \beta_a > Rate$; otherwise, R_{P2P} equals 0. We intend to compute R_{P2P} at one time slot ahead when $R_{P2P} \neq 0$.

However, we also want to maximize the CDN Resource Utilization Ratio (*CRU*). From the perspective of CDN operators, the higher the *CRU,* the more profit gained under the special economic model, in which only the traffic produced by CDN servers must charge. Thus, the optimization problem can present as follows:

$$Max: \quad \sum_{a \in A} CRU * q_c^a \tag{1}$$

subject to

$$CRU = \begin{cases} \sum_{a \in A} n_c^a * \beta_a / Rate, \ if \ \sum_{a \in A} n_c^a * \beta_a \leq Rate \\ (\sum_{a \in A} n_c^a * \beta_a - R_{p2p}) / Rate, \ else \end{cases} \tag{2}$$

$$(\sum_{a \in A} n_c^a * \beta_a - R_{P2P}) \leq Rate \tag{3}$$

$$q_c^a = F^a(n_c^a, S_c^a, R_{P2P}) \tag{4}$$

$$1 \geq q_c^a \geq 0 \tag{5}$$

The objective function in Equation (1) reflects the system goal of maximizing the CDN resource utilization *CRU* and the response ratio to end users' requests q_c^a simultaneously. It is different from the traditional system's goal [4][5][36], in which the only task is to maximize q_c^a and diminish server load through the peer-assisted method. Thus, the situation may occur, such that q_c^a reaches its maximum value while *CRU* is close to be 0. Equation (2) defines *CRU* considering two conditions presented by equation (3) and equation (4). If the aggregated resource demand $\sum_{a \in A} n_c^a * \beta_a$ exceeds the service capacity *Rate*, $\sum_{a \in A} n_c^a * \beta_a - R_{p2p}$ is the total load of CDN servers and it does not exceed *Rate* as shown in Equation (3), which indicates the effectiveness of P2P in assisting CDN. Otherwise, the total load of CDN servers is $\sum_{a \in A} n_c^a * \beta_a$. In equation (4), the service quality function F^a presents the relationship among q_c^a, n_c^a, S_c^a, and R_{P2P}. Since the inherent relationship among them is beyond the scope of our paper, they are encapsulated by F^a and we only need to know q_c^a has complexes relationship with n_c^a, S_c^a, and R_{P2P}. Equation (4) presents the upper bound of q_c^a is 1, which indicates that all the users' requests have been satisfied.

The first challenge in achieving the goal is to answer when and how many server loads must be offloaded, i.e., computing the optimal value of R_{P2P} one time slot ahead. The difficulty here lies in the fluctuating number of end users that will try to access simultaneously multiple applications in the future. The second challenge is the task of designing a more effective and elastic P2P policy to provide moderated peer-assisted content delivery. In the next section, we present our method to solve these challenges.

4. DMPM

We propose the DMPM, which involves the CDN dynamic load online prediction that can identify when and how many end-users requests offloaded through P2P networks. It also has a novel popularity-aware P2P policy that used to cut down the load of the exceeding servers moderately. DMPM conducted on a geo-distributed CDN node without any central control.

4.1 CDN Dynamic Load Online Prediction

We used the time series analysis method [24][25] to estimate resource demand $d_c(t+1)$ at future time $t+1$ inside node *c*, while $d_c(t)$ equals $\sum_{a \in A} n_c^a * \beta_a$ at time *t*. A previous work [26] has predicted end-users behavior in real-world P2P live streaming systems, another study [27] modeled end-users behavior using Poisson and Pareto distribution. Unlike any other end-user behavior mentioned above, we consider in the present study the aggregated behaviors of web, live streaming, VoD, file downloading, and social networks users, among others. Moreover, we predict the end-users aggregation behavior (i.e., the aggregated number of all kinds of end users' requests) $d_c(t+1)$ one-time slot ahead, which refers to online prediction of the dynamic loads of CDN nodes. We treat the random variables collected over time, which is the aggregated resource demand in CDN node (e.g. $d_c(t), t = 1, 2, ...,$) as a time series, and use the recent historical values to forecast the most likely values in the future.

According to our pre-analysis of the time series data of China Cache [28], in terms of resource demand, the linear time series analysis fits our problem. However, we are uncertain as to whether or not the time series of the aggregated resource demand is stationary. Thus, we utilized ARIMA(p,d,q) [24][25]. If the time series is stationary, we only need to set the parameter *d=0*, which implies no *differencing* process and that ARIMA(p,0,q) is the same as the Autoregressive and Moving Average (ARMA)model [24][25]. If the time series is not stationary, we only need *d* times *differential operations* to derive a stationary series, and then use ARMA(p,q) to model the *differenced stationary* series. In these models, each variable in the series can express as the linear weighted sum of *p* previous values and *q* previous random errors. In ARMA(p,q), ARMA(0,q) equals Moving Average (MA) model MA(q),and ARMA(p,0) equals Autoregressive (AR) model AR(p). Thus, ARIMA(p,d,q) is a general series model that is more suitable to our problem. Using ARIMA to realize the online prediction of a CDN's dynamic load involves four steps: (1) stationary test, (2) model identification, (3) model estimation, and (4) prediction.

Algorithm1. *Dynamic Resource Demand Prediction*

Line1. **Input:** The time series $\{d_c(1), d_c(2), \dots, d_c(t)\}$
Line2. **Output:** The predicted value $d_c(t+i)$, $i=\{1,3,5\}$
Line3. // within 5-step ahead prediction is accurate for *ARIMA*.
Line4. **Initialization:** $i=1$; $d=0$; *p* threshold *p_th*=6; *q* threshold *q_th*= 6;
$V_{BIC}[p\_th][q\_th]=\{inf,\dots,inf\}$.
Line5. //Building *Box-Jenkins* Model [13];

Line6. While (there exists *unit root*);//Testing for *Unit Root*;
Line7. {
Line8. One Time Differential Operation;
Line9. $d=d+1$;
Line10. } //Differential Stationary Time Series Got
Line11. for (j=(0:p_th))
Line12. {
Line13. for (k=(0:q_th))
Line14. {
Line15. Computing *ARMA(p,q)*;
Line16. Extracting the Bayesian Information Criterion (BIC) [13] information $V_{BIC}[j,k]$ of *ARMA(j,k)*;
Line17. }
Line18. }
Line19. Search for the minimum value $V_{BIC}[j,k]$ in matrix V_{BIC};
Line20. Set $p=j$, $q=k$;
Line21. Using *ML/ULS/CLS* [13] method to do parameter estimation;
Line22. Using *Significance test* to determine whether *ARMA(p,q)* is significant;
Line23. Using *ARIMA(p,d,q)* [13] to forecast $d_c(t+i)$
Line24. Computing the residual terms;//the data series of prediction error
Line25. While (the residual term series does not pass the *white noise test*)
Line26. {
Line27. *Re-model identification*, *estimation, forecast* et al.
Line28. }

This is a distributed algorithm running on each CDN node. We implemented the algorithm and the time to run the algorithm is millisecond orders of magnitude. The first step is to use the *unit roots test* method to verify whether the time series is stationary (from lines 6 to 10). If it is not stationary, then d times of difference operations shall be operate until a differential stationary data series achieved. Then, we use ARMA(p,q) to model the differential stationary series (from lines 11 to 18). The p and q thresholds are always set to 6, because the data series usually becomes differential stationary after undergoing difference operations twice. As p, d, and q are determined using the Bayesian Information Criterion (BIC) principle (from lines 19 to 20), parameter estimation (line 21), model significance (line 22), forecasting (line 23), and residual term series (line 24), the *white nose test* (from lines 25 to 28) should be performed. The output of the algorithm is the predicted value $d_c(t+i)$, i ($i \in \{1,\ldots,5\}$) steps ahead.

It is important to predict end-users' uncertain requests one time slot ahead, because the Round-trip Time (RTT) is commonly used as the indicator of the distance between two peers when constructing a P2P overlay before a neighbor selection [29]. Thus, the time before the P2P overlay can effectively offload the CDN servers load by at least two RTTs, including selection neighbor delay and connection latency. As discussed previously [30] on DSL or cable internet connections, latencies of less than 100 milliseconds are typical, and less than 25 milliseconds than the desired latency. To improve CDN scalability in time, we need to predict when and how many server loads must be offloaded using P2P technology. In addition, one time slot prediction ahead is enough, regardless of how the RTT value dynamically changes.

4.2 Popularity-Aware P2P Policy

We first set a reasonable threshold Warning Value (*WV*) on the load of each CDN node based on experience, e.g., 85% of the CDN node's Maximum Service Capacity (*MSC*). If end-users' demand does not exceed the threshold *WV*, all end–users are served by the CDN servers using a client/server pattern. Once the server load exceeds the threshold, the CDN begins to predict whether the end users' aggregated requests will exceed the *MSC* of the CDN node in the next time slot. In this case, the number of requests from the end-users (or the resource demand) that exceeds the service capacity of the CDN node should compute. Once the exceeding part has been computed one time slot ahead, we can use the peer-assisted method to deal with these exceeded requirements immediately instead of using the CDN servers. This move can improve CDN scalability, which involves transferring the overhead part of the CDN load to the end-users. It is one of the most popular ways to cut down the load of CDN servers and improve their scalability to assure end users' QoS. As for the method of organizing a large number of end-users to form a P2P overlay, we considered using the existing VUD method [20]. There are two kinds of architectures: hybrid and pure P2P architectures [31]. In this study, we adopt the subclass of the hybrid P2P architecture, in which special CDN servers, also referred to as super peers, are treated as centralized control points for P2P overlay formation, content indexing and searching, peer resource finding, and neighbor list constructing, among others. Thus, once we can predict when and how many CDN server loads need to be offloaded through the P2P content delivery method, we can assign the appropriate number of end-users to form the P2P overlay using the VUD method [20] and achieve the goal of improving CDN scalability moderately and maximizing *CRU*.

The algorithm for improving CDN scalability moderately using peer-assisted manner is presented in Algorithm 2.

Algorithm2. *Popularity-Aware P2P Overlay Construction*

Input: Reasonable threshold *WV* about the loads of each CDN node.
Output: *S*calability *C*oefficient (*SC*).
Initialization: *MSC*, β_a, β_b.
Real-time collection of end-users resource demand; //CDN server load;
If (CDN server load <*WV*)
  {
    Do nothing;
  }
Else
{
    Call Dynamic Resource Demand Prediction and get the predicted $d_c(t+i)$ ahead;
    If ($d_c(t+i)$>*MSC*)
      {
        UED=$d_c(t+i)$-*MSC;//UED is short for End-users' Exceeded Demand for resource.*
        UN=UED/β_a *; //UN is short for the End-users Number, which will be satisfied by P2P;*
        //β_a *is the resource demand of each end user in average;*
        Sorting the contents according to their popularity into a *P*opular *C*ontent *L*ist (*PCL*);
        // the high popularity means the more requests from End-users;
        While (*UN*>0)
          {
            Selecting the *T*op *P*opular *C*ontent (*TPC*);
            Finding the *S*eed *N*odes (*SN*) of *TPC*;
            Computing the capacity (C_{p2p}) that can be offered for *SN* using *VUD* method;
            Forming a P2Poverlay (in which *SN* are seeds) to deliver *TPC* using *VUD* method;
            UN=UN-C_{p2p}/β_b*;//*β_b *is the upload capacity of each seed on average;*
            Deleting *TPC* from *PCL*;
          }
      }
    Computing the *SC*:

```
Line30.    SC=(UED+MSC)/MSC;
Line31.  }
```

This distributed algorithm is similar to that in Algorithm 1. Once the CDN server loads exceed *WV,* the prediction procedure begins (from lines 5 to 9 in Algorithm 2). If the predicted value $d_c(t+i)$ exceeds *MSC* (lines 11 and 12), the exceeding part must be computed exactly (lines 14 and 15), e.g., the number of end-users and the type of application that end-users want to access, and some P2P overlays must be constructed to satisfy the exceeding end-users requests (line 24). As for constructing the P2P overlay, the end-users who access the popular content (line 21) in the near past form the P2P overlay using the VUD method. Many end-users may exist and act as seeds (lines 22 and 23) for requests coming from new end-users. If the CDN has no ability to satisfy the exceeding resource demand from end-users, the exceeding resource demand can be satisfied through the peer-assisted method, indicating the improvement of CDN scalability. However, only the exceeding part uses in the peer-assisted method, which is an advantage for both CDN operators and end-users. As a result, end users' QoS can assure, and the economic interests of CDN operators influenced. Two metrics, namely, SC and CDN economic profit, are guaranteed and improved; moreover, high *SC* does not influence CDN operators' economic interests.

5. Performance Evaluation and Results Analysis

The previous sections presented the problem formulation and the corresponding algorithms. In this section, the performance metrics and evaluation analysis are discussed, with the aim of illustrating the efficiency achievement of the DMPM.

5.1 Performance metrics

In the current paper, we used four performance metrics to evaluate the effectiveness of DMPM, namely, *PE, CRU, RRR*, and *SC*.

1) *PE*

PE denotes the *dynamic resource demand prediction* accuracy for DMPM; a lower *PE* indicates a more accurate prediction method. The accuracy of the prediction indicates that the *dynamic resource demand prediction* method provides CDN operators with a more accurate direction.

$$PE = \frac{\left|n_c(t+i) - \hat{n}_c(t+i)\right|}{\left|n_c(t+i)\right|}, \quad t = 1, 2, \ldots, n \tag{6}$$

In Equation (6), $\hat{n}_c(t+i)$ denotes the predicted value of the aggregated resource demand at time t+i, and $n_c(t+i)$ indicates the real value of the aggregated resource demand at time t+i. Here, $\left|n_c(t+i) - \hat{n}_c(t+i)\right|$ is the absolute prediction error, which also implies the prediction accuracy.

2) *CRU*

CDN operators deploy some rated bandwidth resources in each geo-distributed CDN server node. CDN operators aim to maximize the resources they deploy, no matter how fluctuant the aggregated resource demand is; therefore, even the *CRU* does not reach 100%. CDN operators aim to serve end-users requests using their own CDN server resources. However, when the aggregated resource demand exceeds the CDN's rated service capacity, operators aim only for the exceeding resource demand from end-users, which is satisfied using the peer-assisted method.

$$CRU = \begin{cases} n_c(t+i) / Rate, & if\ n_c(t+i) < Rate \\ (n_c(t+i) - C_{p2p}) / Rate, & else \end{cases} \tag{7}$$

In Equation (7), when $n_c(t+i)$ is more than $Rate$, the approximate CRU value is $n_c(t+i) - C_{p2p}$, which may be a little greater than $Rate$ at some point. The reason is that C_{p2p} depends on the prediction accuracy and the effectiveness of the *VUD* policy, with C_{p2p} being equal to the difference value between $\hat{n}_c(t+i)$ and *Rate*.

3) *RRR*

End-users' *RRR* reflects their QoS. In the context of this paper, end-users requests are responded to, indicating that there are enough resources that can be allocated to the end-users' requests.

$$RRR = \begin{cases} 1, & if\ \ n_c(t+i) < Rate\ \&\ \hat{n}_c(t+i) > n_c(t+i) \\ 1 - (n_c(t+i) - \hat{n}_c(t+i)) / Rate, & the\ others \end{cases} \tag{8}$$

In the case of $\hat{n}_c(t+i) > n_c(t+i)$ in Equation (8), the performance metrics *RRR* is not influenced, but the performance metrics *CRU* is less than 100%. The *RRR* performance metrics is one of the main targets of the CDN operators.

4) *SC*

The service capacity of CDN is limited. Dealing with end-users requests when the CDN does not have enough capacity to supply end-users with services is the main scalability problem of CDN. The *SC* performance metric denotes the scale up ability of CDN.

$$SC = \begin{cases} \hat{n}_c(t+i) / Rate, & if\ \ \hat{n}_c(t+i) > Rate \\ 0, & the\ others \end{cases} \tag{9}$$

In Equation (9), *SC* defined as the predicted value, which is the basic direction of the P2P overlay construction. *SC* represents CDN system service ability, i.e. the service ability of CDN to exceed the rated service capacity using the peer-assisted method.

5.2 Performance Evaluation Analysis

In this section, we present details about the performance evaluation, which was based on the traces collected from a real CDN system, i.e. China Cache [28]. The continuous seven-day traces collected in September 2010 from about 95 geo-distributed CDN server nodes used in China. The distribution of these CDN server nodes is similar to that of the Live Sky server nodes in China for the 17th CPC National Congress [5][32]. The traces indicate the aggregated end users' requests. End-users cover all kinds of applications, such as Web, streaming, and social networks. The seven-day aggregated requests coming from end-users are dynamically fluctuant and used as the case to evaluate the DMPM. In the evaluation, in order to facilitate our expression, all data normalized, e.g., *Rate* equals one.

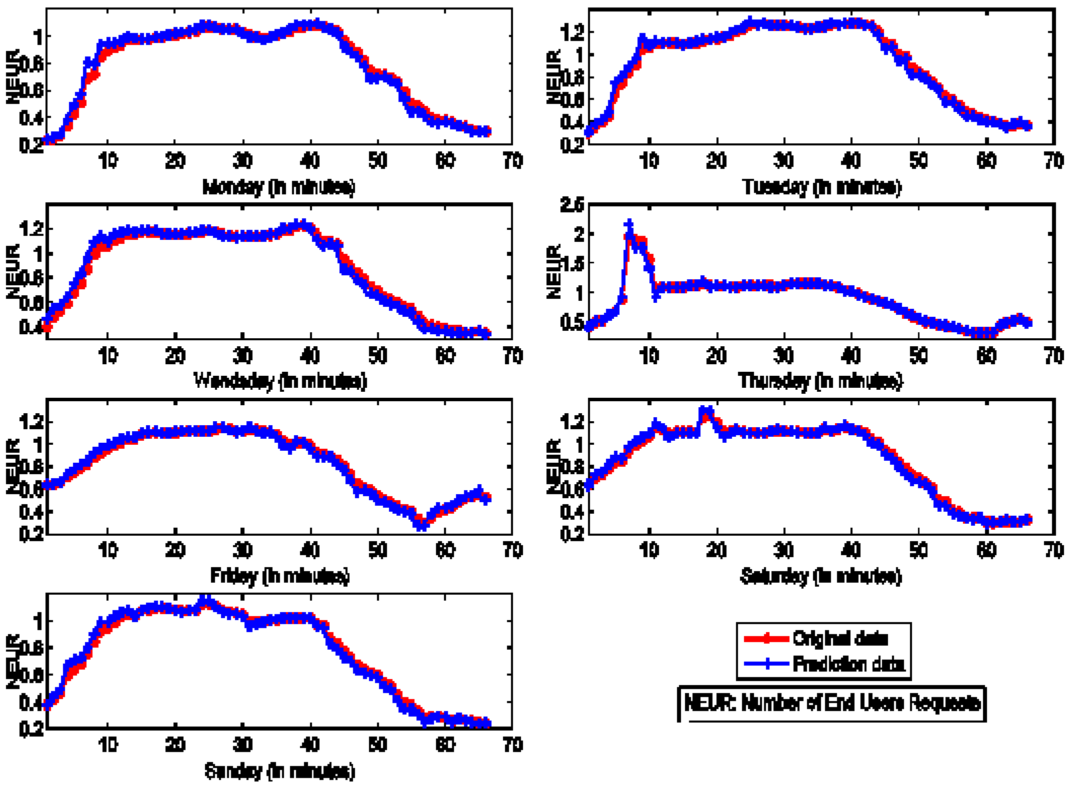


Fig. 3. The Original and Predicted Data from Monday to Sunday.

Fig. 3 presents the original data of the aggregated end users' requests and their predicted data. The original data collected from a real CDN entity for one week, while the predicted data obtained through DMPM. In the course of using DMPM to obtain the predicted data of each day in the week, we use some original data collected immediately before the week as the training data. In **Fig. 3** we only present the predicted data using DMPM and their corresponding original data, which also can be treated as the testing data. Further, analyses about the original and predicted data displayed. Some patterns of end users' requests from Monday to Sunday obtained. From **Fig. 3**, we can see that, the end users' aggregated requests fluctuant daily and that the end users' aggregated requests usually exceed the CDN service capacity for some time during daytime. The exceeding situation leads to the rejection of the end-users' requests by CDN servers, which means that part of their requests cannot be satisfied and part of end-users would not obtain services with guaranteed QoS. However, the exceeding situation is not exactly the same from Monday to Sunday within any given week. For example, on Thursday, the end users' aggregated requests present a sudden increasing phenomenon. The highest number of requests is nearly double the service capacity, which is similar to the bursting resource demand in cloud [2]. In other days, the aggregated resource demand generally exceeds—albeit slightly—the service capacity of CDN and last for some time, e.g. on Tuesday, Wednesday, Friday, and Saturday. Only on Monday and Sunday in this week, would the aggregated resource demand from end-users be lower than the service capacity of CDN. Of course, the data of other weeks may not be the same as those for this week, but the data in the week still have sufficient representativeness to some extent. From **Fig.**

3, we can see that the predicted data are close to the original data. The prediction accuracy presented as follows.

Fig.4 shows the absolute residual values between the original and predicted values. The absolute residual value is equal to the difference between the original data and corresponding predicted value. It can also reflect the prediction accuracy to some extent, e.g. the residual value of 0 refers to the prediction accuracy of 100%。 The distribution of the absolute residual values is different among the seven days. For example, the residual value on Friday is smaller than that on other days, in which the maximum of the residual value is not more than 0.15, except on Thursday in Fig.4.The average residual absolute value is about 0.05, except on Thursday. For Thursday, the maximum residual value has reached nearly 2.5, which means the prediction method used in DMPM is not fit to predict the data with a pattern like that found on Thursday. Obviously, which kind of data fits the prediction using our method is still an open problem.

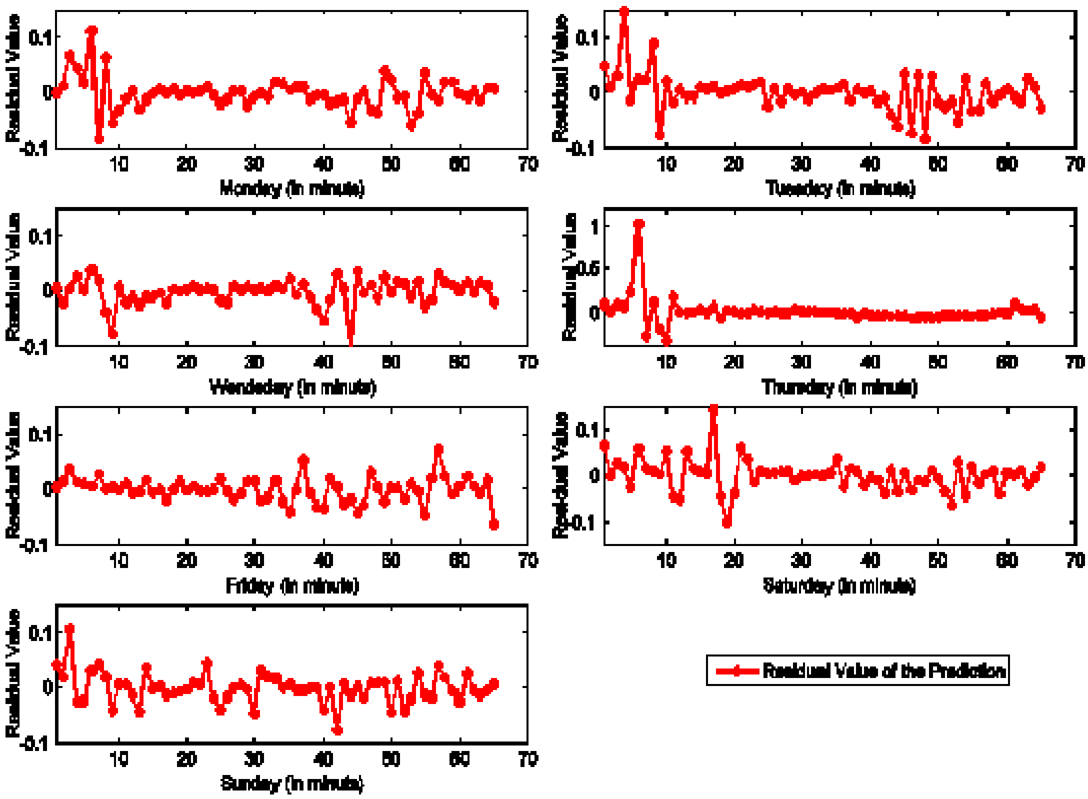


Fig. 4. The Residual Value of the Prediction From Monday to Sunday.

The residual values can partly reflect the prediction accuracy of the ARIMA model used in our DMPM. Furthermore, *PE* used to measure the exact effectiveness of the ARIMA model in predicting the dynamic resource demand. Fig. 5 indicates that most of the predicted values in *PE* are less than 0.1. Except on Thursday, during which the minimum *PE* value is higher than 0.2. This finding indicates that the ARIMA model is not suitable for any dynamic resource demand prediction. Thus, our practice in verifying the original data in the week except on Thursday is very significant. In Fig. 5, the distribution of the performance metric *CRU* indicates that after using DMPM, the CDN resource utilization maximized. Maximizing CDN resource utilization is one of the most important targets for the CDN operators. When the end users' aggregated request does not exceed the service capacity of the CDN, the CDN server nodes, thus implying that the resource utilization cannot be higher, serve all requests. When the aggregated request exceeds the service capacity of the CDN, only the exceeding part uses the P2P method moderately. Hence, the CDN resource utilization is able to maintain at almost 100% under this circumstance.

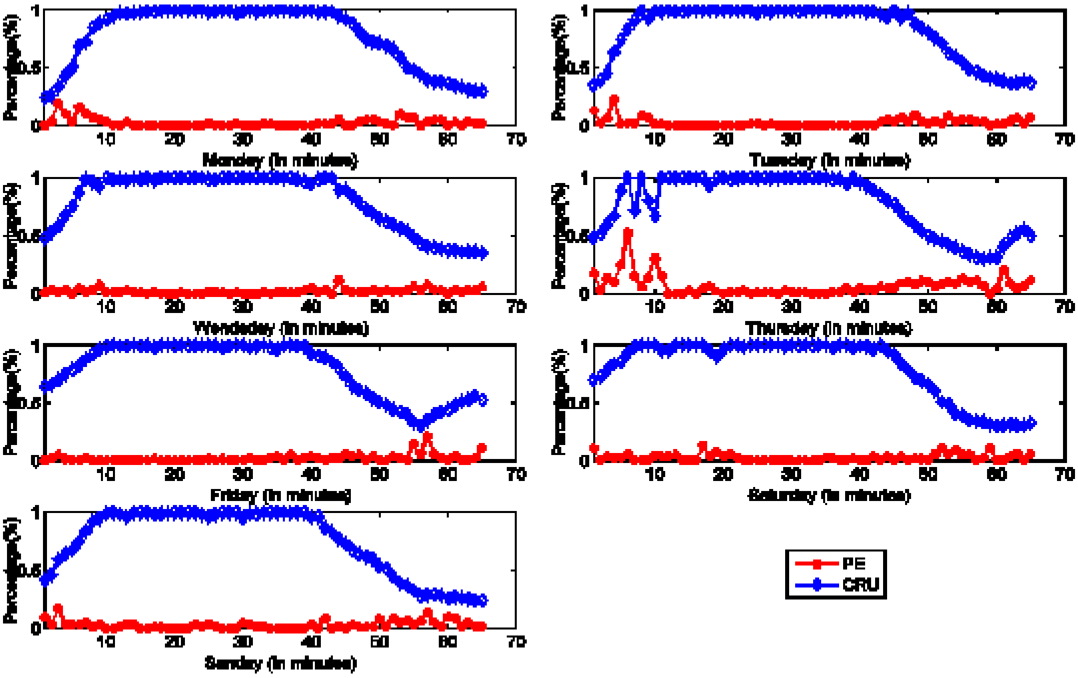


Fig. 5. The PE and CRU Distribution from Monday to Sunday.

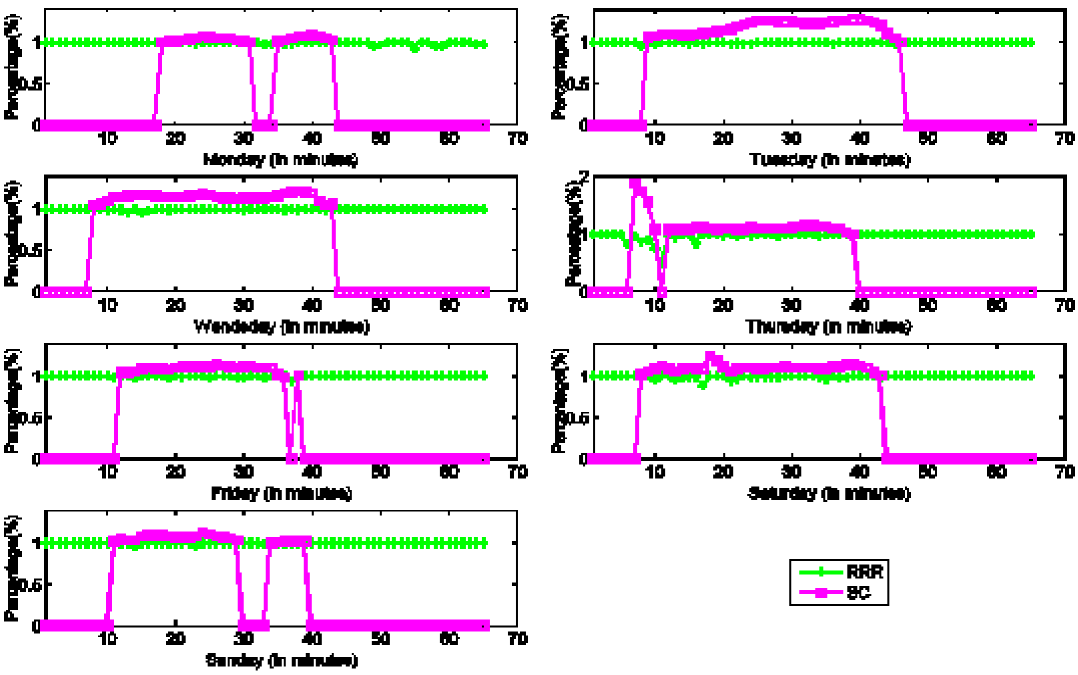


Fig. 6. The Distribution of *RRR* and *SC* from Monday to Sunday.

Fig. 6 presents the distribution of the two performance metrics *RRR* and *SC*. After using DMPM, we can see that *RRR* is always equal to one, implying that all end-users requests can be accommodated CDN server nodes or the P2P overlay (**Fig. 6**), as *RRR* present the QoS of end-users. Only on Thursday, the *RRR* appears lower than one. Due to the burst phenomenon influence, many end users' requests cannot get guaranteed QoS. In the context of this paper, end-users requests are responded to, indicating that there are enough CDN server resources or P2P resources that can be allocated to respond to the end-users' requests. *RRR* approaching one means almost all of the end-users obtained services with guaranteed QoS. Based on the *SC* distribution shown in **Fig. 6**, CDN scalability increases along with the increase in end users' aggregated requests. This implies that regardless of changes in the resource demand from the end-users, the service capacity of CDN can catch up. However, the CDN only needs to scale up when the aggregated end users' requests exceed the CDN's service capacity. With *SC* equals to zero in **Fig.6**, it means that the aggregated end users' request is lower than the service capacity of CDN and the later does not need to scale up its service capacity. In comparison, when the *SC* is higher than one in **Fig. 6**, this means that DMPM started and P2P has improved the service capacity of CDN.

Based on the above-mentioned analysis, we conclude that after using DMPM, the CDN service scalability can improve moderately and the maximization of the end users' QoS is guaranteed.

6. Conclusion

To improve CDN scalability moderately based on the perspective of CND operators, i.e., maximizing CDN resource utilization and guaranteeing the QoS of end-users at the same time, the current paper presents an effective method called DMPM. First, we proposed a system model and presented the problem formulation. Subsequently, we presented the DMPM, which helps solve "when and how to use P2P in CDN to improve CDN scalability moderately through the peer-assisted method". In DMPM, the ARIMA model used to predict and decide when and how many CDN server loads must cut down when CDN do not have enough available resources to satisfy end-users requests. Furthermore, a novel P2P mechanism, such as the VUD, adopted to assist CDN as it improves CDN's scalability. Finally, real traces from China Cache used to evaluate DMPM and prove the effectiveness. The results imply that the combination of the ARIMA prediction model and P2P technology used as a mechanism design for the improvement of CDN scalability.

References

[1] Lucas Mearian, "World's data will grow by 50X in next decade, IDC study predicts," 2011.

[2] M. Armbrust, A.Fox,R. Griffith, A. Joseph, R. Katz, A.Konwinski, G. Lee, D.Patternson, A.Rabkin, I.Stoica and M. Zaharia, "A view of cloud computing," *Communications of ACM*, vol.53, no.4, pp.50-58, 2010. Article (CrossRef Link)

[3] M. Begam and G. Ganapathy, "A Cloud- 'Network becomes super computer'," in *Proc. of The 2nd Conference on Technology Management*, pp.64-75, 2011.

[4] D. Xu, S.Kulkarni, C. Rosenberg and H. Chai, "Analysis of a CDN-P2P hybrid architecture for cost-effective streaming distribution," *Multimedia Systems,* vol.1, no.4,pp.383-399, 2006. Article (CrossRef Link)

[5] H. Yin, X. Liu, T. Zhan, V.Sekar, F.Qiu,C. Lin, H.Zhang and B. Li, "Design and deployment of a hybrid CDN-P2P system for live video streaming: Experiences with livesky," in *Proc. of ACM Multimedia*, pp.25-34, 2009. Article (CrossRef Link)

[6] C. Huang, A. Wang, J. Li and K. Ross, "Understanding hybrid CDN-P2P: why limelight needs its own Red Swoosh," in *Proc. of 18th International workshop on Network and Operating Systems Support for Digital Audio and Video*, pp.75-80, 2008. Article (CrossRef Link)
[7] T. Karagiannis, P. Rodriguez and K. Papagiannaki, "Should internet service providers fear Peer-Assisted content distribution?" in *Proc. of Internet Measurement Conference*, pp.63-76, 2005. Article (CrossRef Link)
[8] D. Pakkala and J. Latvakoski, "Towards a Peer-to-Peer extended content delivery network," in *Proc. of 14th IST Mobile and Wireless Communications Summit*, pp.1-5, 2005.
[9] P. Rodriguez, S. Tan, C. Gkantsidis, "On the feasibility of Commercial, LegalP2P Content Distribution," *ACM SIGCOMM Computer Communication Revie*w, vol.36, no.1, pp.75-78, 2006. Article (CrossRef Link)
[10] J. Tirado, D. Higuero, F. Isaila, J.Carretero and A. Iamnitchi, "Affinity P2P: A self-organizing content-based locality-aware collaborative peer-to-peer network," *Computer Networks*, vol.54, no.12, pp.2056-2070, 2010. Article (CrossRef Link)
[11] I. Stoica, R. Morris, D. Karger, M. Kaashoek and H. Balakrishnan, "Chord: a scalable peer-to-peer lookup service for internet applications," in *Proc. of SIGCOMM,* pp.149-160, 2001. Article (CrossRef Link)
[12] A.Rowstron, P. Druschel, "Pastry: Scalable, decentralized object location, and routing for large-scale Peer-to-peer systems," *Lecture Notes in Computer Science*, vol.2218, no.2001, pp.329-350, 2001. Article (CrossRef Link)
[13] D. Li, X. Lu and J. Su, "Graph-Theoretic analysis of Kautz topology and DHT schemes," *Lecture Notes in Computer Science*, vol.3222, no.2004, pp.308-315, 2004. Article (CrossRef Link)
[14] D. Li, J. Cao, X. Lu and K. Chen, "Efficient range query processing in Peer-to-Peer systems," *IEEE Transactions on Knowledge and Data Engineering*, vol.21, no.1, pp.78-91, 2009. Article (CrossRef Link)
[15] Bolla R and Gaeta R, Magnetto A, Sciuto M and Sereno M, "A measurement study supporting P2P File-sharing community models," *Computer Networks*, vol.53, no.2009, pp.485-500, 2009. Article (CrossRef Link)
[16] I. Clarke, O. Sandberg, B. Wiley and T. Hong, "Freenet: a distributed anonymous information storage and retrieval system," in *Proc. of Workshop on Design Issues in Anonymity and Unobservability*, pp.46-56, 2001.
[17] Y. Huang, T. Fu, D. Chiu, J.Lui and C. Huang, "Challenges, design and analysis of a large-scale P2P-vod system," in *Proc. of SICOMM*, pp.375-388, 2008. Article (CrossRef Link)
[18] T. Bonald, L. Massoulie, F. Mathieu, D. Perino and A. Twigg, "Epidemic live streaming: Optimal performance trade-offs," in *Proc. of ACM SIGMETRICS*, pp.325-336, 2008. Article (CrossRef Link)
[19] S. Liu, S. Zhang, W. Jiang, J. Rexford and M. Chiang, "Performance bounds for Peer-Assisted live dtreaming," in *Proc. of ACM international conference on Measurement and modeling of computer systems*, pp. 313-324, 2008. Article (CrossRef Link)
[20] D. Wu, Y. Liu and K. Ross, "Queuing network models for Multi-Channel P2P Live Streaming Systems," in *Proc. of INFOCOMM*, pp.73-81, 2009. Article (CrossRef Link)
[21] M. Castro, P. Druschel, A. Kermarrec, Nandi A, A. Rowstron and A. Singh, "SplitStream: High-bandwidth multicast in cooperative environments," in *Proc. of ACM SOSP*, pp.298-313, 2003.
[22] N. Magharei and R. Rejaie, "PRIME: Peer-to-Peer Receiver-driven Mesh-based Streaming," in *Proc. of INFOCOM*, pp. 1415-1423, 2007. Article (CrossRef Link)
[23] M. Zhang, J. Luo, L. Zhao and S. Yang, "A Peer-to-peer network for live media sreaming using a Push-pull approach," in *Proc. of ACM Multimedia,* pp.287-290, 2005. Article (CrossRef Link)
[24] G. Box, G. Jenkins and G. Reinsel, "Time series analysis: Forecasting and control (Third Edition)," *Prentice Hall*, 1994.
[25] R. Tsay, "Analysis of financial time series."*John Wiley: New York*, 2002. Article (CrossRef Link)
[26] C. Wu, B. Li and S. Zhao, "Multi-channel Live P2P streaming: Refocusing on servers," in *Proc. of INFOCOM*, pp.1355-1363, 2008. Article (CrossRef Link)

[27] R. Kumar, Y. Liu and K. Ross, "Stochastic fluid theory for P2P streaming systems," in *Proc. of INFOCOM*, pp.919-927, 2007. Article (CrossRef Link)
[28] ChinaCache Inc., "http://www.chinacache.com/," 2011.
[29] A. Russo and R.Cigno, "Delay-Aware Push/Pull protocols for live video streaming in P2P systems," in *Proc. of IEEE International Conference on Communications,* pp.1-5, 2010. Article (CrossRef Link)
[30] B.Mitchell, "Network Bandwidth and Latency," , http://compnetworking.about.com/od/speedtests/a/network_latency.htm
[31] T. Oh-ishi, K. Sakai, K. Kikuma and A. Kurokawa, "Study of the relationship between Peer-to-peer systems and IP multicasting," *IEEE Communication Magazine*, vol.41, no.1, pp.80-84, 2003. Article (CrossRef Link)
[32] H. Yin, X. Liu, T. Zhan, V.Sekar, F. Qiu, C. Lin, H. Zhang and B. Li, "LiveSky: Enhancing CDN with P2P," *ACM Transaction on Multimedia Computing, Communications, and Applications,* vol.6, no.3, pp.1-19, 2010. Article (CrossRef Link)
[33] D. Niu, Z. Liu, B. Li and S. Zhao, "Demand forecast and performance prediction in peer-Assisted On-Demand streaming systems," in *Proc. of INFOCOM*, pp.421-425, 2011. Article (CrossRef Link)
[34] S. Islam, J. Keung, K. Lee and A. Liu, "Empirical prediction models for adaptive resource provisioning in the cloud," *Future Generation Computer Systems*, vol.28, no.1, pp.155-162, 2012. Article (CrossRef Link)
[35] Chow-Sing Lin, "Improving the availability of scalable on-demand streams by dynamic buffering on P2P networks," *KSII TIIS Journal,* vol.4, no.4, pp.491-508. Article (CrossRef Link)
[36] Kim-Thinh Nguyen and Young-Han Kim, "A CDN-P2P hybrid architecture with location content awareness for live Streaming services," *KSII TIIS Journal,* vol.5, no.11, pp.2143-2159. Article(CrossRefLink)
[37] Xicheng Lu, Huaimin Wang, Ji Wang, Jie Xu and Dongsheng Li, "Internet-based virtual computing environment: beyond the data center as a computer," *Future Generation Computer Systems*, Aug.2011. Article (CrossRef Link)

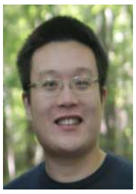
Peichang Shi is a Ph.D candidate in National University of Defense Technology. He received a B.E. degree in Information Security from Wuhan University, and a M.E. in Cryptography from National University of Defense Technology. His research interests include Information Security, Cloud Computing, Content Distribution Networks etc.

Huaimin Wang is a professor in the School of Computer Science of National University of Defense Technology. He received his Ph.D. in Computer Science from National University of Defense Technology (NUDT) in 1992. He is now a professor and vice dean in the College of Computer, NUDT. He has been awarded the "Chang Jiang Scholars Program" professor by Ministry of Education of China, and the Distinct Young Scholar by the National Natural Science Foundation of China (NSFC), etc. He has worked as the director of several grand research projects and has published more than 100 research papers in international conferences and journals. His current research interests include middleware, software Agent, trustworthy computing

Yin Hao is a research fellow in the Department of Computer Science of Tsinghua University. He received a B.E. degree and Ph.D. degree from Huazhong University of Science and Technology, and as a postdoctoral fellow in Tsinghua University. His current research is in Content Delivery Networks, Multimedia Communication and Information Security etc.

Bo Ding is an assistant research fellow in the School of Computer Science of National University of Defense Technology. He received a M.S. degree and Ph.D. degree from National University of Defense Technology. His current research is in Trustworthy Computing, Pervasive Computing.

Tianzuo Wang is a Ph.D candidate in National University of Defense Technology. He received the B.S. and M.S. from National University of Defense Technology. His research interests include distributed computing and information security



Miao Wang is currently a research engineer at A10Networks, San Jose, CA. He obtained Ph.D degree majoring in Computer Science at the University of Nebraska-Lincoln (UNL) in May 2011. He received his B.E. and M.E. degrees in Computer Science from Xi'an Jiaotong University in 2003 and 2006 respectively. His research interest spans on Peer-to-Peer computing, multimedia streaming and network congestion control

.