Improving Content Delivery by Exploiting the Utility of CDN Servers

George Pallis

Computer Science Department
University of Cyprus
gpallis@cs.ucy.ac.cy

Abstract. As more aspects of our work and life move online and the Internet expands beyond a communication medium to become a platform for business and society, Content Delivery Networks (CDNs) have recently gained momentum in the Internet computing landscape. Today, a large portion of Internet traffic is originating from CDNs. The ultimate success of CDNs requires novel policies that would address the increasing demand for content. Here we exploit the CDN utility - a metric that captures the traffic activity in a CDN, expressing the usefulness of surrogate servers in terms of data circulation in the network. We address the content replication problem by replicating content across a geographically distributed set of servers and redirect users to the closest server in terms of CDN utility. Through a detailed simulation environment, using real and synthetically generated data sets we show that the proposed approach can improve significantly the performance of Internet-based content delivery.

Keywords: Internet-based Content Delivery, Replication, CDN pricing.

1 Introduction

Content Delivery Networks (CDNs) have emerged to overcome the inherent limitations of the Internet in terms of user perceived Quality of Service (QoS) when accessing Web data. They offer infrastructure and mechanisms to deliver content and services in a scalable manner, and enhance users' Web experience. Specifically, a CDN is an overlay network across the Internet, which consists of a set of servers (distributed around the world), routers and network elements. Edge servers are the key elements in a CDN, acting as proxy caches that serve directly cached content to users. With CDNs, content is distributed to edge cache servers located close to users, resulting in fast, reliable applications and Web services for the users. Once a user requests content on a Web provider (managed by a CDN), the user's request is directed to the appropriate CDN server. The perceived high end-user performance and cost savings of using CDNs have already urged many Web entrepreneurs to make contracts with CDNs. For instance, Akamai - one of the largest CDN providers in the world - claims to be delivering 20% of the world's Web traffic. While the real numbers are debatable, it is clear that CDNs play a crucial role in the modern Internet infrastructure.

A. Hameurlain et al. (Eds.): Globe 2012, LNCS 7450, pp. 88–99, 2012.

In [10], we introduced a metric, called CDN utility metric, in order to measure the utility of CDN servers. A similar metric has also been used in [12] for a p2p system. Recent works [8,9,13] have shown that CDN utility metric captures the traffic activity in a CDN, expressing the usefulness of CDN servers in terms of data circulation in the network. In particular, the CDN utility is a metric that expresses the relation between the number of bytes of the served content against the number of bytes of the pulled content (from origin or other CDN servers). In this work we go one step further and propose a utility-based replication policy for CDNs. Maximizing the CDN utility metric, we can both reduce network utilization and overall traffic volume, which may further reduce network congestion and thus improve the overall user performance. Using an extensive simulation testbed, we show that this metric can improve the CDN's performance. Our contributions in this paper can be summarized as follows:

- We formulate the problem of optimally replicating the outsourced content in surrogate servers of a CDN.
- We present a novel adaptive technique under a CDN framework on which we replicate content across a geographically distributed set of servers and redirect users to the closest server in terms of CDN utility. Maximizing the CDN utility, we maximize the usefulness of CDN surrogate servers and minimize the final cost of Web content providers.
- We develop an analytic simulation environment to test the efficiency of the proposed scheme. Using real and synthetically generated data sets, we show the robustness and efficiency of the proposed method which can reap performance benefits.

In this work, we use the CDN utility metric which has been introduced in [10], in order to determine the placement of outsourced content to CDN surrogate servers. Considering that CDN utility expresses the relation between the number of bytes of the served content against the number of bytes of the pulled content (from origin or other surrogate servers), maximizing this value would improve content delivery.

The structure of the paper is as follows. In Section 2, we formulate the content replication problem for improving content delivery and in Section 3 the proposed approach is described. Sections 4 and 5 present the simulation testbed and the evaluation results respectively. Section 6 presents an overview of related work and Section 7 concludes the paper.

2 Problem Formulation

In CDNs, the content is pushed (proactively) from the origin Web server to CDN surrogate servers and then, the surrogate servers cooperate in order to reduce the replication and update cost.

We consider a popular Web site that signs a contract with a CDN provider with N surrogate servers, each of which acts as an intermediary between the servers and the end-users. We further assume that the surrogate server i has S_i bytes of storage capacity, where $i \in \{1, ..., N\}$.

In order to formulate the placements cost function, we assume that we have K requested objects. Each object k has a size of s_k, where $k \in \{1, ..., K\}$. In this context, we define a variable which determines if an object k is stored to surrogate server i as follows:

$$f_{ik} = 1 \tag{1}$$

if object k is stored at surrogate i, else $f_{ik} = 0$.

The storage is subject to the constraint that the space available at surrogate server i is bounded by $\sum_{k=1}^{K} s_k f_{ik} \leq S_i$, where $i \in \{1, ..., N\}$. Furthermore, each surrogate server can hold at most one replica of the object.

Considering that all the requested objects are initially placed on an origin server (the initial placement is denoted by x_o), the content replication problem is to select the optimal placement x (defines the placement of requested objects to CDN surrogate servers) such that it minimizes:

$$cost(x) = \sum_{i=1}^{N} \sum_{k=1}^{K} p_{ik}(D_{ik}(x)) \tag{2}$$

where $D_{ik}(x)$ is the "distance" to a replica of object k from surrogate server i under the placement x and p_{ik} is the likelihood that the object k will be requested by server i. This likelihood encapsulates the users' satisfaction.

However, as it has been proved in previous studies, this problem is NP complete (it is similar to the NP-complete knapsack problem), which means that for a large number of requested objects and surrogate servers is not feasible to solve this problem optimally. In this context, we propose a new heuristic strategy where its criterion is the CDN utility metric.

3 The CDN Utility Replica Placement Algorithm

The main idea is to place the requested objects to surrogate servers with respect to CDN utility metric. The intuition of this metric is that a surrogate server is considered to be useful (high net utility) if it uploads content more than it downloads, and vice versa. It is bounded to the range [0..1] and provides an indication about the CDN traffic activity.

Formally, the CDN utility u is expressed as the mean of the individual utilities of each surrogate server [10]. Considering that a CDN has N surrogate servers, the CDN utility u can be defined as follows:

$$u = \frac{\sum_{i=1}^{N} u_i}{N} \tag{3}$$

where u_i is the net utility of a CDN surrogate server i and it is quantified by using the following equation:

$$u_i = \frac{2}{\pi} \times arctan(\xi) \tag{4}$$

Algorithm 1. The CDN Utility Replica Placement Algorithm

Input: $O[1..K]$: users' requested objects;
Input: $S[1..N]$: CDN surrogate servers;
Output: a placement x of requested objects to surrogate servers

```
1: for all obj ∈ O do
2:    for all s ∈ S do
3:       compute the CDN-utility[obj][s];
4:    end for
5: end for
6: for all obj ∈ O do
7:    for all s ∈ S do
8:       create a list L: each element of the list includes the pairs of outsoursed object
         surrogate server ordering with respect to the maximum CDN-utility;
9:    end for
10: end for
11: while (list L not empty AND cache size of S not full) do
12:    get the first element from L: e=(s-obj, s-s);
13:    if (size of s-obj ≤ cache size of s-s) then
14:       place s-obj to s-s;
15:    end if
16:    delete e from L;
17: end while
```

Fig. 1. The CDN Utility Replica Placement Algorithm

The parameter ξ is the ratio of the uploaded bytes to the downloaded bytes. The arctan function in (4) assists to obtain scaled resulting utility in the range [0..1]. The value $u_i = 1$ is achieved if the surrogate server uploads only content ($\xi = infinity$). On the contrary, the value 0 is achieved if the surrogate server downloads only content. In the case of equal upload and download, the resulting value is 0.5.

Initially all the requested objects are stored in the origin server and all the CDN surrogate servers are empty. Taking into account the users requests, for each requested object, we find which is the best surrogate server in order to place it. In our context, the best surrogate server is the one that produces the maximum CDN utility value.

Then, we select from all the pairs of requested objectsurrogate server that have been occurred in the previous step, the one which produces the largest CDN utility value, and thus place this object to that surrogate server. The intuition of this placement is to maximize the utility of CDN surrogate servers. The above process is iterated until all the surrogate servers become full. As a result, a requested object may be assigned to several surrogate servers, but a surrogate server will have at maximum one copy of a requested object. The detailed algorithm is described in pseudocode in Figure 1. From the above, it occurs that the *utility* approach is heavily dependent on the incoming traffic. Specifically, previous research [9] has shown that the number of CDN surrogate

servers, network proximity, congestion and traffic load of servers impact the resulting utility in a CDN. Concerning the complexity of the proposed algorithm is polynomial, since each phase requires polynomial time. In order to by-pass this problem, we may use content clusters (e.g., Web page communities) [4].

CDN Pricing. The CDN utility replica placement algorithm would be beneficial for CDN pricing. Specifically, we integrate the notion of net utility in the pricing model that has been presented in [1]. In particular, the monetary cost of the Web content provider under a CDN infrastructure is determined by the following equation:

$$U_{CDN} = V(X) + \tau(N) \times X - C_o - P(u) \tag{5}$$

where U_{CDN} is the final cost of Web content provider under a CDN infrastructure, $V(X)$ is the benefit of the content provider by responding to the whole request volume X, $\tau(N)$ is the benefit per request from faster content delivery through a geographically distributed set of N CDN surrogate servers, C_o is cost of outsourcing content delivery, $P(u)$ is the usage-based pricing function, and u is the CDN utility. Consequently, maximizing the CDN utility, $P(u)$ is also maximized and according to equation 5 the final cost of Web content provider under a CDN infrastructure (U_{CDN}) is minimized.

4 Simulation Testbed

CDN providers are real-time applications and they are not used for research purposes. Therefore, for the evaluation purposes, it is crucial to have a simulation testbed for the CDN functionalities and the Internet topology. Furthermore, we need a collection of Web users traces which access a Web server content through a CDN, as well as, the topology of this Web server content (in order to identify the Web page communities). Although we can find several users traces on the Web, real traces from CDN providers are not available. Thus, we are faced to use artificial data. Regarding the Web content providers, we use three real data sets. The data sets come from an active popular news Web content provider (BBC). Also, we take into account that the lifetime of Web objects is short. The expired Web objects are removed from the cache of surrogate servers. Table 1 presents the data sets used in the experiments.

This work is in line with the simulation-based evaluation of utility as described in [10]. We have developed a full simulation environment, which includes the following:

- a system model simulating the CDN infrastructure,
- a network topology generator,
- a client request stream generator capturing the main characteristics of Web users behavior.

Table 1. Summary of simulations parameters

Web site	BBC
Web site size	$568MB$
Web site number of objects	17000
Number of requests	1000000
Mean interarrival time of the requests	$1sec$
Distribution of the interarrival time	exponential
Requests stream z	0.5
Link speed	$1Gbps$
Network topology backbone type	AS
Number of routers in network backbone	3037
Number of surrogate servers	100
Number of client groups	100
Number of content providers	1
Cache size percentage of the Web site's size	6.25%, 12.5%, 25%, 50%

4.1 CDN Model

To evaluate the proposed approach, we used our complete simulation environment, called CDNsim [11], which simulates a main CDN infrastructure and is implemented in the C programming language. A demo can be found at http://oswinds.csd.auth.gr/∼cdnsim/. It is based on the OMNeT++ library[1] which provides a discrete event simulation environment. All CDN networking issues, like CDN servers selection, propagation, queueing, bottlenecks and processing delays are computed dynamically via CDNsim, which provides a detailed implementation of the TCP/IP protocol (and HTTP), implementing packet switching, packet re-transmission upon misses, objects' freshness etc.

By default, CDNsim simulates a CDN with 100 CDN servers which have been located all over the world. Each CDN server in CDNsim is configured to support 1000 simultaneous connections. The default size of each surrogate server has been defined as the percentage of the total bytes of the Web server content. We also consider that each surrogate server cache is updated using a standard LRU cache replacement policy.

4.2 Network Topology

In a CDN topology we may identify the following network elements: CDN surrogate servers, origin server (Content Provider's main server), routers and clients. The routers form the network backbone where the rest of the network elements are attached. The distribution of servers and clients in the network affects the performance of the CDN. Different network backbone types result in different "neighborhoods" of the network elements. Therefore, the redirection of the requests and ultimately the distribution of the content is affected. In our testbed we use the AS Internet topology. The routers retransmit network packets using the TCP/IP protocol between the clients and the CDN. All the network phenomena such as bottlenecks and network delays, and packet routing protocols are simulated. Note that the AS Internet topology with a total of 3037 nodes captures a realistic Internet topology by using BGP routing data collected from a set of 7 geographically-dispersed BGP peers. In order to minimize the side effects due to intense network traffic we assume a high performance network with 1 Gbps link speed.

[1] `http://www.omnetpp.org/article.php?story=20080208111358100`

In order to ground our model in reality we have parameterized our CDN infrastructure with the real properties of a commercial CDN provider (Limelight). According to a recent study by [2][2], Limelight has clusters of servers deployed at 19 different locations around the world and each cluster has a different number of servers. In the United States there are 10 clusters, whereas, in Europe and Asia are seven clusters in total and Australia has only one. The rest of the world does not contain any cluster.

4.3 Requests Generation

As far as the requests stream generation is concerned, we used a generator, which reflects quite well the real users access patterns. Specifically, this generator, given a Web site graph, generates transactions as sequences of page traversals (random walks) upon the site graph, by modelling the Zipfian distribution to pages. In this work, we have generated 1 million users' requests. Each request is for a single object, unless it contains "embedded" objects. According to Zipfs law, the higher the value of z is the smaller portion of objects covers the majority of the requests. For instance, if z = 0 then all the objects have equal probability to be requested. If z = 1 then the probability of the objects fade exponentially. In this work we used the range 0.5 for z in order to capture an average case.

Then, the Web users' requests are assigned to CDN surrogate servers taking into account the network proximity and the surrogate servers' load, which is the typical way followed by CDN providers. Specifically, the following CDN redirection policy takes place: 1) A client performs a request for an object. 2) The request is redirected transparently to the closest surrogate server A in terms of network topology distance. 3) The surrogate server uploads the object, if it is stored in its cache. Elsewhere, the request is redirected to the closest to A surrogate server B, that contains the object. Then, the surrogate server A downloads the object from B and places it in its cache. If the object cannot be served by the CDN, the surrogate server A downloads the object from the origin server directly. 4) Finally the object is uploaded to the client.

5 Evaluation

In order to evaluate the proposed algorithm, we examine also the following heuristics: a) *Lat-cdn*: The outsourced objects are placed to surrogate servers with respect to the total network latency, without taking into account the objects popularity. Specifically, each surrogate server stores the outsourced objects which produce the maximum latency [7]; b) *il2p*: the outsourced objects are placed to the surrogate servers with respect to the total network latency and the objects load [6].

We evaluate the performance of CDN under regular traffic. It should be noted that for all the experiments we have a warm-up phase for the surrogate servers

[2] The paper has been withdrawn by Microsoft. We only use information about server locations from this work.

caches. The purpose of the warm-up phase is to allow the surrogate servers caches to reach some level of stability and it is not evaluated. The measures used in the experiments are considered to be the most indicative ones for performance evaluation.

5.1 Evaluation Measures

CDN Utility: It is the mean of the individual net utilities of each surrogate server in a CDN. The net utility is the normalized ratio of uploaded bytes to downloaded bytes. Thus, the CDN utility ranges in [0..1]. Using the notion of CDN utility we express the traffic activity in the entire CDN. Values over 0.5 indicate that the CDN uploads more content than it downloads through cooperation with other surrogate servers or the origin server. In particular, for the uploaded bytes we consider the content uploaded to the clients and to the surrogate servers. Values lower than 0.5 are not expected in the CDN schemes. The value 0.5 is an extreme case where each request is an object that has not been served by CDN.

Hit Ratio: It is the ratio of requests that has been served, without cooperation with other surrogate servers or the origin server, to the total number of requests. It ranges in [0..1]. High values of hit ratio are desired since they lead to reduced response times and reduced cooperation. Usually the hit ratio is improved by increasing the cache size and it is affected by the cache replacement policy.

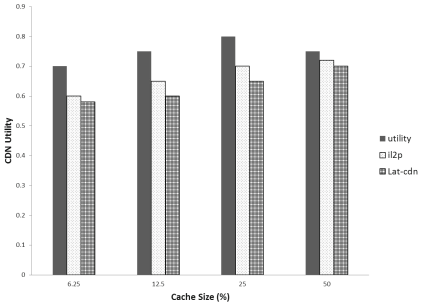
Mean Response Time: It is the mean of the serving times of the requests to the clients. This metric expresses the users experience by the use of CDN. Lower values indicate fast served content.

Finally, we use the *t-test* to assess the reliability of the experiments. T-test is a significance test that can measure results effectiveness. In particular, the t-test would provide us an evidence whether the observed mean response time is due to chance. When a simulation testbed is used for performance evaluation, it is critical to provide evidence that the observed difference in effectiveness is not due to chance. For this purpose, we conducted a series of experiments, making random permutation of users (keeping the rest of the parameters unchanged).

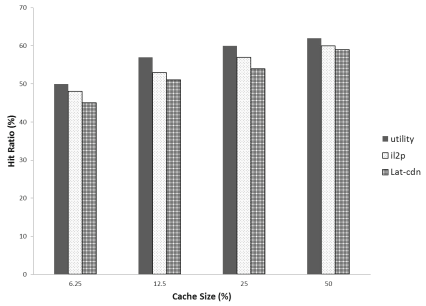
5.2 Evaluation Results

The results are reported in Figures 2(a), 2(b) and 2(c), where x-axis represents the different values of cache size as a percentage of the total size of distinct objects (total Web site size).

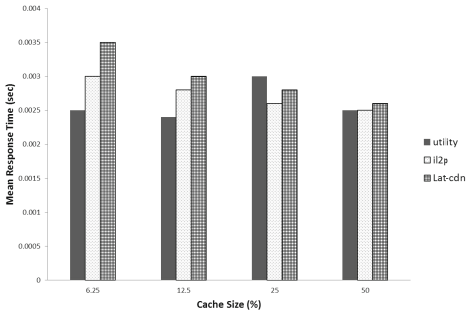
Figure 2(a) depicts the CDN utility evolution that can be achieved by the CDN utility replica placement algorithm for different cache sizes. A general observation is that the proposed algorithm achieves high CDN utility values for any cache size comparing with the competitive ones. Taking a deeper look at the results, we observe a peak in the performance of the CDN utility at 25% cache size. This means that the *utility* approach achieves low data redundancy. Giving more insight to the performance peak, we should identify what happens to the



(a) CDN Utility vs Cache Size.



(b) Hit Ratio vs Cache Size.



(c) Mean Response time vs Cache Size.

Fig. 2. Evaluation results

eras before and after the peak. Before the peak, the cache size is quite small. Few replicas are outsourced to the CDN servers and the surrogate servers fail to cooperate. Most of the requests refer to objects that are not outsourced at all. Consequently, the surrogate servers refer to the origin server in order to gain copies of these objects. This leads to lower CDN utility as the surrogate servers upload less content. As the cache size increases, the amount of replicated content in the CDN increases as well. Therefore, the cooperation among the surrogate servers is now feasible and thus the CDN utility increases. After the peak, the cache size is large enough; consequently, this results in reducing the cooperation among the surrogate servers.

Figure 2(b) presents the hit ratio of the proposed algorithm for different values of cache sizes. As expected, the larger the cache size is, the larger the hit ratio is. Results show that the *utility* approach outperforms the other two examined policies. The most notable observation is that the proposed replication algorithm achieves high hit ratio for small cache sizes. Replicating a small size of Web site content, the observed performance peak guarantees satisfactory performance by reducing the traffic to origin server.

Figure 2(c) depicts the mean response time for different cache sizes. The general trend for the *utility* algorithm is that the response time is low for any

cache size, observing for the *utility* approach a peak of mean response time at 25% cache size. The explanation for this peak is due to the fact that more requests are served; note that we also observe a peak in the performance of the CDN utility at 25% cache size (see Figure 2(a)).

Regarding the statistical test analysis, the t statistic is used to test the following null hypothesis (H_0):

H_0: The observed mean response time and the expected mean response time are significantly different.

We computed the $p - value$ of the test, which is the probability of getting a value of the test statistic as extreme as or more extreme than that observed by chance alone, if the null hypothesis H_0, is true. The smaller the $p - value$ is, the more convincing is the rejection of the null hypothesis. Specifically, we found that $p - value < 0,001$, which provides a strong evidence that the observed MRT is not due to chance.

6 Related Work

The growing interest in CDNs is motivated by a common problem across disciplines: how does one reduce the load on the origin server and the traffic on the Internet, and ultimately improve content delivery? In this direction, crucial data management issues should be addressed. A very important issue is the optimal placement of the requested content to CDN servers. Taking into account that this problem is NP complete, an heuristic method should be developed.

A number of research efforts have investigated the problem of content replication problem. Authors in [3] conclude that Greedy-Global heuristic algorithms are the best choice in making the replication. Replicating content across a geographically distributed set of servers and redirecting clients to the closest server in terms of latency has emerged as a common paradigm for improving client performance [5]. In [7], a self-tuning, parameterless algorithm, called lat-cdn, for optimally placing requested objects in CDNs surrogate servers is presented. Lat-cdn is based on network latency (an objects latency is defined as the delay between a request for a Web object and receiving that object in its entirety). The main advantage of this algorithm is that it does not require popularity statistics, since the use of them has often several drawbacks (e.g. quite a long time to collect reliable request statistics, the popularity of each object varies considerably etc.). However, this approach does not take into consideration the load of the objects (the load of an object is defined as the product of its access rate and size). Therefore, in this approach, it is possible to replicate in the same surrogate server objects with high loads and, thus, during a flash crowd event the server will be overloaded. To address this limitation, another approach is presented in [6] for optimally placing outsourced objects in CDN surrogate servers, integrating both the networks latency and the objects load.

However, all the existing approaches do not consider the CDN utility as a parameter to decide where to replicate the outsourced content. This is a key measure for CDNs since the notion of CDN utility can be used as a parameter to

CDN pricing policy. Typically, a CDN outsources content on behalf of content provider and charges according to a usage (traffic) based pricing function. In this context, several works exist which present the utility computing notion for content delivery [8,9,10]. They mostly provide description of architecture, system features and challenges related to the design and development of a utility computing platform for CDNs.

7 Conclusion

In this paper, we addressed the problem of optimally replicating the outsourced content in surrogate servers of a CDN. Under a CDN infrastructure (with a given set of surrogate servers) and a chosen content for delivery it is crucial to determine in which surrogate servers the outsourced content should be replicated. Differently from all other relevant heuristics approaches, we used the CDN utility metric in order to determine in which surrogate servers to place the outsourced objects. Summarizing the results, we made the following conclusions:

- The proposed algorithm achieves low replica redundancy. This is very important for CDN providers since low replica redundancy reduces the computing and network resources required for the content to remain updated. Furthermore, it reduces the bandwidth requirements for Web servers content, which is important economically for both individual Web servers content and for the CDN provider itself.
- The proposed algorithm achieves a performance peak in terms of CDN utility at a certain small cache size. Considering that the capacity allocation in surrogate servers affects the pricing of CDN providers we view this finding as particular important.

The experiment results are quite encouraging to further investigate the CDN utility replica placement algorithm under different traffic types and simulation parameters.

Acknowledgement. We are indebted to Theodoros Demetriou for his feedback on the early drafts of this work. This work was supported by the author's Startup Grant, funded by the University of Cyprus.

References

1. Hosanagar, K., Chuang, J., Krishnan, R., Smith, M.D.: Service adoption and pricing of content delivery network (cdn) services. Manage. Sci. 54, 1579–1593 (2008)
2. Huang, C., Wang, A., Li, J., Ross, K.W.: Measuring and evaluating large-scale cdns paper withdrawn at mirosoft's request. In: Proceedings of the 8th ACM SIGCOMM Conference on Internet Measurement, IMC 2008, pp. 15–29. ACM, New York (2008)
3. Kangasharju, J., Roberts, J., France Tlcom, R., Ross, K.W., Antipolis, S., Antipolis, S.: Object replication strategies in content distribution networks. Computer Communications, 367–383 (2001)

4. Katsaros, D., Pallis, G., Stamos, K., Vakali, A., Sidiropoulos, A., Manolopoulos, Y.: Cdns content outsourcing via generalized communities. IEEE Transactions on Knowledge and Data Engineering 21(1), 137–151 (2009)
5. Krishnan, R., Madhyastha, H.V., Srinivasan, S., Jain, S., Krishnamurthy, A., Anderson, T., Gao, J.: Moving beyond end-to-end path information to optimize cdn performance. In: Proceedings of the 9th ACM SIGCOMM Conference on Internet Measurement Conference, IMC 2009, pp. 190–201. ACM, New York (2009)
6. Pallis, G., Stamos, K., Vakali, A., Katsaros, D., Sidiropoulos, A., Manolopoulos, Y.: Replication based on objects load under a content distribution network. In: 22nd International Conference on Data Engineering Workshops, p. 53 (2006)
7. Pallis, G., Vakali, A., Stamos, K., Sidiropoulos, A., Katsaros, D., Manolopoulos, Y.: A latency-based object placement approach in content distribution networks. Web Congress, Latin American, 140–147 (2005)
8. Pathan, M., Broberg, J., Buyya, R.: Maximizing Utility for Content Delivery Clouds. In: Vossen, G., Long, D.D.E., Yu, J.X. (eds.) WISE 2009. LNCS, vol. 5802, pp. 13–28. Springer, Heidelberg (2009)
9. Pathan, M., Buyya, R.: A utility model for peering of multi-provider content delivery services. In: IEEE 34th Conference on Local Computer Networks, LCN 2009, pp. 475–482 (October 2009)
10. Stamos, K., Pallis, G., Vakali, A., Dikaiakos, M.D.: Evaluating the utility of content delivery networks. In: Proceedings of the 4th Edition of the UPGRADE-CN Workshop on Use of P2P, GRID and Agents for the Development of Content Networks, UPGRADE-CN 2009, pp. 11–20. ACM, New York (2009)
11. Stamos, K., Pallis, G., Vakali, A., Katsaros, D., Sidiropoulos, A., Manolopoulos, Y.: Cdnsim: A simulation tool for content distribution networks. ACM Trans. Model. Comput. Simul. 20, 10:1–10:40 (2010)
12. Tangpong, A., Kesidis, G.: A simple reputation model for bittorrent-like incentives. In: International Conference on Game Theory for Networks, GameNets 2009, pp. 603–610 (May 2009)
13. ul Islam, S., Stamos, K., Pierson, J.-M., Vakali, A.: Utilization-Aware Redirection Policy in CDN: A Case for Energy Conservation. In: Kranzlmüller, D., Toja, A.M. (eds.) ICT-GLOW 2011. LNCS, vol. 6868, pp. 180–187. Springer, Heidelberg (2011)