Proceedings of the 2016 Winter Simulation Conference
T. M. K. Roeder, P. I. Frazier, R. Szechtman, E. Zhou, T. Huschka, and S. E. Chick, eds.

SIMULATION AND OPTIMIZATION OF CONTENT DELIVERY NETWORKS CONSIDERING USER PROFILES AND PREFERENCES OF INTERNET SERVICE PROVIDERS

Peter Hillmann
Tobias Uhlig
Gabi Dreo Rodosek
Oliver Rose

Department of Computer Science
Universität der Bundeswehr München
Werner-Heisenberg-Weg 39
Neubiberg, 85577, GERMANY

ABSTRACT

A Content Delivery Network (CDN) is a dynamic and complex service system. It causes a huge amount of traffic on the network infrastructure of Internet Service Providers (ISPs). Oftentimes, CDN providers and ISPs struggle to find an efficient and appropriate way to cooperate for mutual benefits. This challenge is key to push the quality of service (QoS) for the end-user. We model, simulate, and optimize the behavior of a CDN to provide cooperative solutions and to improve the QoS. Therefor, we determine reasonable server locations, balance the amount of servers and improve the user assignments to the servers. These aspects influence run time effects like caching at the server, response time and network load at specific links. Especially, user request history and profiles are considered to improve the overall performance. Since we consider multiple objectives, we aim to provide a diverse set of pareto optimal solutions using simulation based optimization.

1 INTRODUCTION

Content Delivery Networks (CDN) are large, distributed systems of servers that provide a storage infrastructure. They represent a network platform to provide data services to end-users with high availability and high performance. Providers of a CDN offer these infrastructures to content providers with intention to deliver their information to end-users.

The servers of the CDN copy and cache the content of the original source, to improve the network and server performance. Effectively, a CDN reduces the need for content providers to own and mange a dedicated distribution infrastructure for their content. We focus on collaborative and community driven CDNs that replicate data based on demand independently of its source. CDNs process large number of requests leading to a high amount of transmission data transferred by the Internet Service Provider (ISP). To avoid bottlenecks, severs should be placed at favorable locations with appropriate network connection. Especially, positioning servers at gateways of providers of other network infrastructures ensures the scalability and reliability of the system. At the same time, the length of the network path from an offering source to the requesting destination should be minimized. Finally, the strategic assignment of requests to nearby and appropriate server can significantly improve the load on the server and the network. From an end-user perspective, we obtain a higher performance and reliable quality.

A collaboration between CDN and ISP is necessary to enable an efficient network platform as a service for content providers and users. The challenge is to set up a cooperation that mutually benefits both parties. This paper introduces a concept for an effective CDN-ISP collaboration. It supports the planning of a CDN

978-1-5090-4486-3/16/$31.00 ©2016 IEEE

including its server locations and storage properties as well as the user assignment at runtime. Our approach is evaluated by simulating the basic processes of a CDN using a given ISP infrastructure. We demonstrate that a cooperation lead to improvements for both parties. Furthermore, we evaluate the behaviors of the users according request profiles based on their request history.

This paper is structured as follows: In Section 2 we describe a typical scenario and the requirements for a CDN-ISP collaboration. Section 3 provide an overview of other approaches in that application area. The main part in Section 4 describes our model of the CDN and the simulation-based optimization approach. Subsequently, we explain the process of optimization, evaluate our concept, and identify fundamental properties of the presented system in Section 5 and 6. The last section summarizes our work and provides and outlook.

2 SCENARIO AND REQUIREMENTS

The management of a CDN intends to improve their infrastructure and service. In consultation with the responsible ISP, a predefined amount of mirror servers has to be placed and connected strategically to the ISP network infrastructure. An example setup is shown in Figure 1, the network of Dialtelecom (Knight et al. 2011). The left side presents the geographical location of the network nodes and the edges of the infrastructure. The right side includes the registered mirror servers. The blue lines represent the assignments of the network nodes to their closest server, i.e., the intended user allocation.

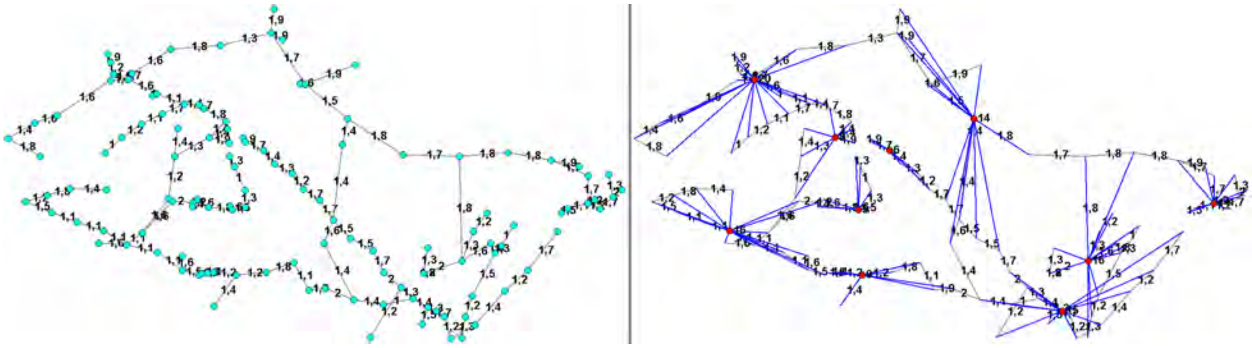

Figure 1: Example of the abstract model with a non-hierarchical network infrastructure including 126 weighted edges and 124 grouped users to 10 prioritized network nodes. In this case, the network nodes are simply assigned to the closest server.

However, the closest server may not be the best one to satisfy a given user request, depending on the available data at the server, the load on the server and the current network performance. With consideration to multiple relevant objectives, the following questions and criteria describe the most common management challenges from CDN providers and ISPs:

- Server-Placement: Where should servers be placed in the network infrastructure? Usually, servers should be placed close to the end-users to improve QoS *(CDN)* and reduce network load as well as lower the risk of bottlenecks *(ISP)*.
- User-Assignment: Which user is assigned to which server? End-users should be assigned to a nearby server with suitable content to get a cache hit and quick response *(CDN)*. This avoids redirecting and minimizes network load caused by data access from another source*(ISP)*. Generally, we may accept larger distances to the servers if we can expect higher cache hit probabilities.
- Server-Amount: What is the necessary amount of servers? An appropriate number of server enables a specific Service-Level with minimal cost of operation *(CDN)* and reduces transition costs *(ISP)*.

- Cache-Size: How much storage do we need for a certain server to obtain an effective system? A suitable amount of caching space improves the storage behavior and balances the operational costs *(CDN)*. It also lowers the amount of redirects to other sources *(ISP)*.

Figure 2 illustrates the dependencies between these aspects. All of them directly or indirectly impact the other ones. For example, the server-amount influences directly the server-placement and the user-assignment. Furthermore, the cache-size influenced indirectly.

Circle of conflict: A higher amount of servers results in an improved placement with shorter network paths. The average connection distances get reduced and QoS is increased. User have access to more nearby servers and thereby more potential assignments. A single server receives fewer requests. The explicit need for large cache sizes is reduced, however, this leads to an increased amount of cache misses. In contrast, a reduced amount of servers will lead to opposite effects.

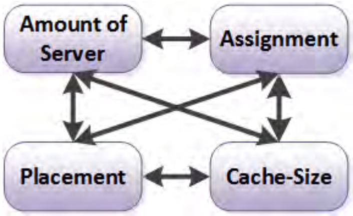

Figure 2: Connection of the different optimization criteria in context of the application area CDN.

The server-placement and the user-assignment are each considered to be NP-hard problems. To find suitable solutions, we need to optimize them simultaneously, because they are strongly connected to each other. The position of a server is influenced by the network infrastructure and the assigned user. Every user should have a minimized connection length to his assigned server. The group of assigned users to a server depends on their user profiles and the network path. The grouped profiles should be positively correlated so that the server obtains requests for the same data. Therefore, the best server position in relation to a short network connection does not necessary result in a group assignment with positive correlated profiles. Furthermore, the adequate amount of server has to be analyzed. Equation 1 and Equation 2 are the objective function for the complex optimization problem, where $user_i$ is assigned to $server_u(user_i)$. To provide a practical tool for CDNs and ISPs, we must determine a whole pareto front of possible solutions to identify the sweet spot for an effective cooperation.

$$Distance = \min \max_{user_i=1,\ldots,n} distance(user_i|server_u(user_i)) \tag{1}$$

$$Correlation = \max \sum_{user_i=1}^{n} \rho(user_i|server_u(user_i)) \tag{2}$$

3 RELATED WORK

Previously, some solutions for an effective CDN-ISP cooperation have been proposed. The work of Pallis and Vakali (2006) presents an overview on CDNs and identifies the most common practices. It conveys a rough idea on balancing costs of CDN providers and content providers while respecting the QoS for Web customers. However, it does not consider the role of ISPs and lacks any detailed suggestions for reaching typical optimization objectives. Nevertheless, their work highlights the importance of appropriate server placement and content management for servers. They, especially, identified dynamic caching-related processes for CDNs as an area with great optimization potential. Mobasher, Cooley, and Srivastava (2000) cluster users according to their profile and the history of URL demands. These clusters are used to predict future HTTP requests. However, their work considers only a single Web Server of a company. We extend the idea of user profiles to a large scale system for a CDN-ISP cooperation to improve the overall performance. The publication of Qiu, Padmanabhan, and Voelker (2001) addresses placement strategies for Web Server replicas in contrast to redirecting requests. They use workload information to save appropriate content on already existing mirror servers to improve access latency and load balance. They also discuss practical issues like imperfect information. Nevertheless, this work is limited to the network aspect of a complex system without taking advantage of user profiles. One of the most influential paper is from Frank et al. (2013). Their concept allows a CDN to incorporate recommendations from an ISP to assign an end-user to a server and to determine locations of servers and content as well as the required amount of servers. The advantage of their system is that no routing changes are necessary to reduce operational costs. Their main focus is on the communication and management processes between CDN Provider and ISP, but not on solving the underlying challenges. They also ignore dynamic problem aspects since they do not use an analytical simulation model. The ALTO protocol (Alimi et al. 2014) is an application layer protocol according to the ISO/OSI seven-layer model. It provides information that are usually hidden, including: network topologies, link availability, routing policies, and path costs from a cooperating ISPs. This information is required to implement and realize our approach in practice.

The work of Jiang et al. (2009) demonstrates that considering server selection and traffic engineering separately leads to sub-optimal solutions. It serves as a starting point to better understand these interaction for those that operate networks and those that distribute content. Their static system does not take user profiles into account. They purely focus on the network costs. Krishnan et al. (2009) investigated latency-based server selection and redirection to improve the CDN performance. This analysis does not consider the effects of dynamic caching behavior at the server and whether requested content is actually available at the specific server. Overall, these approaches used a static system description ignoring the dynamic effects in user behavior driven CDN. Furthermore, they focus more on economic and financial interests without considerations of an improved quality of service for the user. In contrast to the existing work, we analyze the CDN-ISP collaboration with dynamic effects using simulation-based optimization. Therefore we are able to consider run time effects and optimize according to multiple objectives.

4 OPTIMIZATION AND SIMULATION MODEL

Our generic model is based on an existing network infrastructure. We use publicly available data from the *Internet Topology Zoo* (Knight et al. 2011). Nevertheless, the approach is adaptable to any other graph structure. For an effective and reliable service, the mirror servers of the CDN should be connected strategically to the network infrastructure. A server should be placed close to the end-user to provide a continuous service with respect to the network path. This also avoids bottlenecks and queuing delays during transmissions. Different links are weighted according to their properties like bandwidth, latency and utilization. The necessary information is provided by the ISP. Due to the expected high amount of served requests from distributed end-users, the servers need a high-performance connection. This is provided by the backbone networks in the TIER 1 or 2 topology. Therefore, we focus on Top-Level infrastructures to identify optimized mirror server locations. A server is always connected to an existing network node in the given infrastructure. The short link from a server to the selected node is negligible in relation to a long network path to the end-user. Additionally, locations can be prioritized depending on other aspects like operating costs and performance. The resulting placement constraint corresponds to the placement of a mirror server at a weighted node from the backbone topology respecting connection distances. We consider a placement of a server at a data link or in an area without direct connections as impractical.

Usually, the end-users are connected in the TIER 3 topology, which has a tree like structure. Effectively, the users share a common internet gateway, for example one for a street, or one for a region. The necessary structural information has to be provided by the ISP. It includes the geographical location of the end-user connection to the internet gateway as well as the local infrastructure. To reduce the amount of optimization and simulation objects, end-users are aggregated. Multiple end-users connected to the same access node are grouped to represent the respective node. This level of abstraction is precise enough, because every

CDN request from such a group has to pass their common gateway node. All end-user properties and behaviors are also combined. The demands of the users are accumulated to a group profile. To differentiate between groups of end-user, the network node is assigned a priority and a request profile. The priority depends on aspects like the amount of users, their economical relevance, and frequency of requests. The request profile reflects the history and prevalence of requested data and URLs. The required information is usually tracked by the CDN provider to predict future requests and to improve service (Mobasher, Cooley, and Srivastava 2000). We model the end-user profiles using recorded, request traces from file access and cache benchmarks (headissue GmbH 2016). Alternatively, we generate pseudo realistic profiles using the Zipf-distribution (Breslau et al. 1999). Each modeled profile entry consists of the name of service and the probability that the user requests it at a specific time. A profile can have multiple entries.

An example scenario is visualized in Figure 3. It includes prioritized network nodes for groups of users and for placement opportunities of servers. The network nodes are connected with each other via several, weighted links using the existing infrastructure. The requested content is described by the profile of the network node and represented by a pie chart.

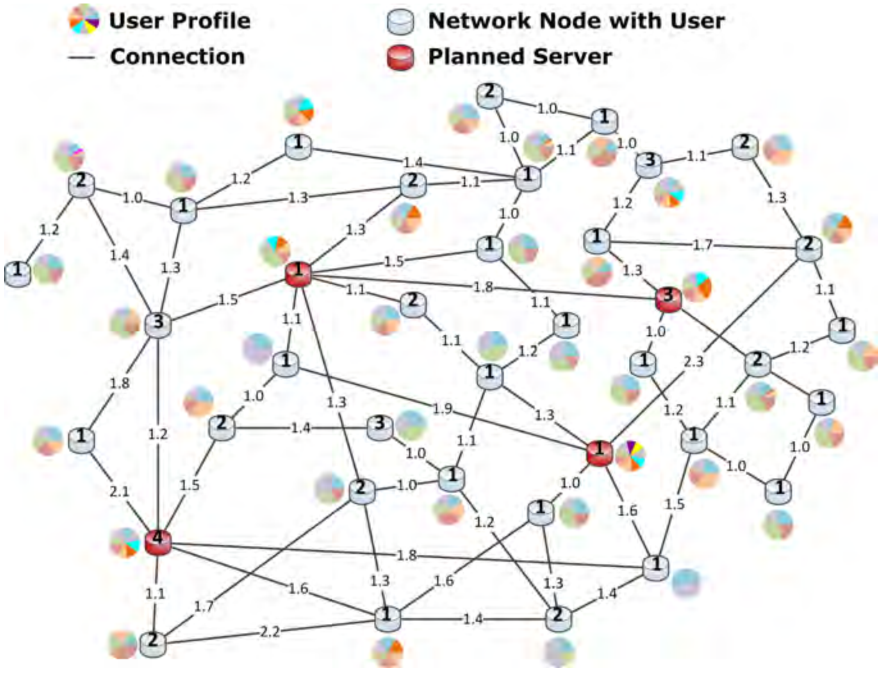

Figure 3: Example of the abstract model with a non-hierarchical network infrastructure including weighted edges and grouped users to a prioritized network node.

During optimization, we place a predefined amount of servers in the network and assign the user groups to their appropriate server. The server placement incorporates multiple aspects. The shortest network path between server and user are calculated with Dijkstra's algorithm. We also factor in priorities, weights and profile correlation. Accordingly, we consider the following optimization objectives: Minimization of maximum or average distance between user and assigned server and profile correlation based on Spearman's rank correlation coefficient.

In our simulation, the end-users send requests to their server. The server is responsible to fully answer the request with the desired data. This specific behavior is dependent on the organization and configuration of the CDN. More complex types with redirecting requests will be considered in future. To answer a request, the necessary data have to be available on that specific server. Otherwise, it is fetched directly from the source or another server in the CDN. In addition to answering the user requests, a server manages its cache and the provision of missing data. To simulate this behavior, we model a cache at servers with a predefined size and cache replacement strategy, e.g., LRU-*X*, LFU, LRFU (Lee et al. 2001), and LIRS (Jiang and Zhang 2002). The cache stores the readily available data and tracks statistical values like miss rate. The optimal caching behavior is calculated according to *Belady* (Belady 1966) afterwards. For simplification purpose, we assume uniform data size. During Simulation we track the network load and the connection distance of the communicating entities. A high QoS is reached by combining a high cache hit ratio and a short transmission path — our main objectives.

The process of model creation, optimization and simulation is presented in Figure 4. The data of the scenario is combined with the placement constraints and enriched with the priorities and weightings of CDN provider and ISP. The created model contains the network infrastructure, locations of users and their demanding profile. Furthermore, it includes information on the specified amount of servers and the specific objectives. During the placement optimization, servers are added to the model and rearranged to improve the performance. During initialization, we consider only static criteria, which can be obtained without simulation. Afterwards, the model is simulated and evaluated. In an iterative process, the server locations and the user assignments are optimized. This is done with respect to the aforementioned static and dynamic optimization objectives. The result includes amongst other things the server locations and the assignment of users to their preferable server.

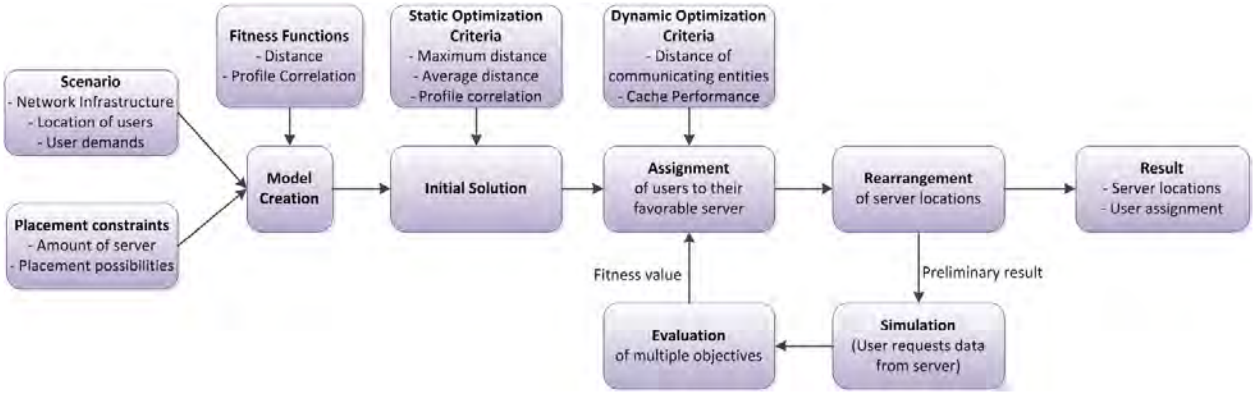

Figure 4: Processing scheme to optimize the placement of mirror server.

5 OPTIMIZATION

For the initial solution, we focus on an optimized server placement without paying attention to the user profiles. It represents the ideal case with a high locality of similar user profiles. Afterwards, the solution is modified stepwise with regard to fitting profiles. The server locations have to be identified with respect to the infrastructure. In intention to equal important users, we minimize the maximum distance from a server to its assigned user. We use the placement algorithms presented in the next section and additionally use an AMOSA algorithm (Bandyopadhyay et al. 2008) as reference.

5.1 Placement Algorithm

We designed an algorithm to find optimized locations for server in a given topology, called Dragoon (Diversification Rectifies Advanced Greedy Overdetermined Optimization N-Dimensions). It uses of two stages: initialization and iterative improvement.

As first step of initialization, an orientation mark is placed at the optimal center position assuming a single center placement problem - calculate all possible constellations for one server. The orientation mark is only used for the placement decision of the first server and discarded afterwards. The first server is placed at the position of a network node which is farthest away from the orientation mark. Subsequently, the remaining number of the pre-defined amount of server is placed using the 2-Approx strategy. It calculates for every network node the distance to all placed server. Afterwards, it chooses the network node with the largest distance to its closest server as the next location to place an additional server. Thereby, we obtain a specific solution of the 2-Approx placement strategy, which originally places the first node randomly.

After the initialization, the algorithm starts with the iterative refinement rule to further improve the server locations. For every server, it tests all connected network nodes from the considered infrastructure with a direct edge to the current position of a server. If the new location improves the overall situation, the algorithm shifts the server to the better position. This is done with respect to the specified optimization objectives.

In each iteration step, all network nodes of the observed infrastructure are (re)assigned to their closest server. Every server is allowed to shift its position only once in each iteration. This iterative optimization is repeated until all server do not change their positions any more. Since the algorithm accepts only improved positions in every step, it is guaranteed to terminate ultimately. These optimized locations for server represent central nodes in a given network topology. Other clustering algorithms like MacQueen (Macqueen 1967), Lloyd (Lloyd 2006), Greedy (Jamin et al. 2001) and Integer Linear Programming (Kleinberg and Tardos 2013) were evaluated and used as references.

5.2 Profile Correlation Algorithm

The spectrum of the user interests is very large. If every service generate a dimension in the multi modal search space, the distance between two profiles become meaningless. So we use a different approach to obtain the Pareto Front starting from optimized placement.

We use an adapted Greedy algorithm to optimize the initial solution towards a profile correlation. It improves the assignment of the users to its preferable server. Therefore, the algorithm focus on the improvement of increased correlation coefficients at every server. The server location is adjusted afterwards. All user profiles assigned to a server are combined to a server profile. A possible reassignment of a user can have a large impact on the correlation. The coefficient represents the fitness of all user requests arriving the same server. So, we include the profile of the currently considered user into the future server profile on a reassignment. For the evaluation, the Spearman's rank coefficient is used as fitness value. The following example explains the calculation of the correlation coefficient. A user profile is comparable with entries in a table, which consists of the name of a service and the probability to be requested. In this example, user 1 has the profile $\{$A=50%, B=50%, C=0%$\}$ and user 2 has the profile $\{$A=30%, B=0%, C= 70%$\}$. Both user are assigned to the same server. The aggregated server profile is $\{$A=40%, B=25%, C=35%$\}$. According to the rang correlation, user 1 obtain a coefficient of ρ=0.125 and user 2 of ρ=0.5.

For every user, the algorithm calculates the correlation coefficient with every server. If the correlation coefficient to the new server is positive, it compares the value with the coefficient to the current server. The user will be reassigned on an improved correlation coefficient at the end of each iteration. All users are simultaneously reassigned to their improved server with the best correlation coefficient. This process is repeated until no reassignment take place. This process will always terminate, as we only accept improved constellations. Other heuristics algorithms like Genetic Algorithms and Simulated Annealing are also adapted to this problem and used for comparison. Furthermore, different correlation coefficients are tested.

6 SIMULATION AND ASSESSMENT

We evaluated our approach based on simulation experiments using scenarios from the Internet Topology Zoo (Knight et al. 2011). The detailed results depend strongly on the specific scenario and and its network infrastructure. Although, we tested many scenarios we will focus our discussion on the example scenario presented in Section 2. Generally we observed comparable results for the other scenarios. With regard to the user profiles, the universe has 100 different services and we used the Zipf's distribution with $\alpha = 0.3$

for profile generation, a typical value to model web traffic. For a simulation run each user sends at least 100 request to its server.

Experiment 1: Initially, we optimized the server placement to a reduced maximum distance from a user to its closest server. During the simulation, every user requests its closest server and has to get the answer from it. Figure 5 show the maximum hop distance for different amount of center nodes, blue and gray line.

With growing numbers of centers, the optimization algorithms Dragoon and Integer Linear Programming (ILP) are steadily able to reduce the maximum distance. Dragoon shows a significantly better performance so it is used for the further evaluation. Additionally, the figure compares the cache miss ratio in percent with different simulation parameter. The cache replacement strategy is calculated according to the optimal decision with *Belady*.

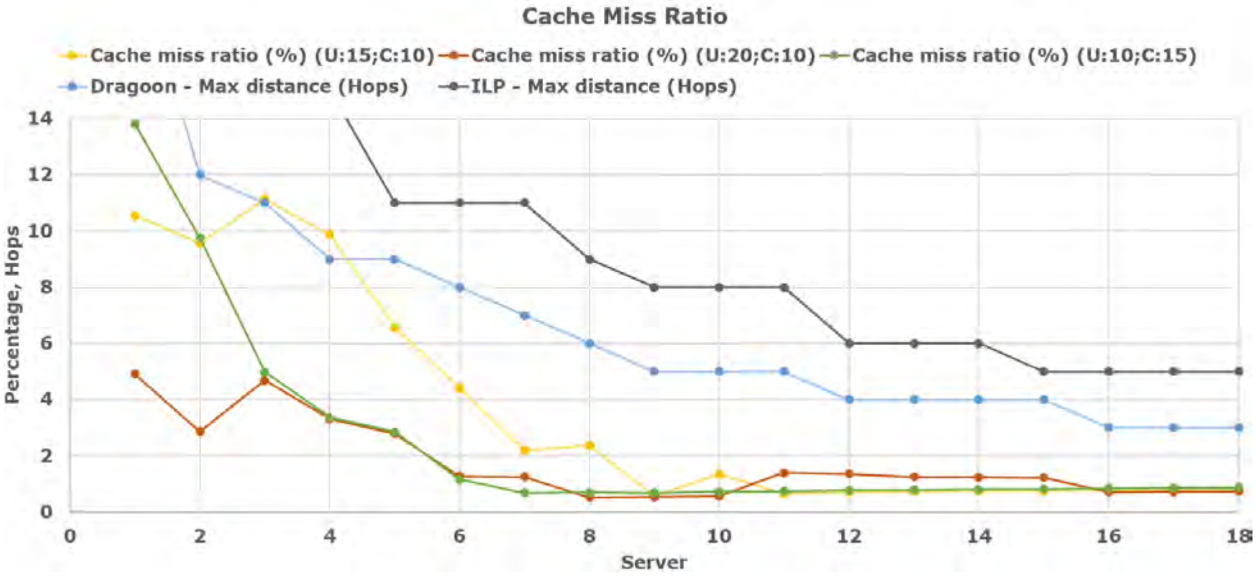

Figure 5: Simulation of the cache miss ratio for different amount of server using the example scenario in Section 2. A user profile consists of U services and a server buffer up to C services.

Generally, a low amount of servers results in a higher cache miss ratio. Nevertheless, not every additional server improves the caching behavior. An additional server can lead to a negative profile correlation of user requests at the server, which increase the cache miss ratio. This effect is heavily visible for low amounts of server and larger user profiles than server caches. The slight increase for large amount of servers can be attributed to a higher number of initial cache misses.

Experiment 2: In this simulation, we improve the correlation of the user profiles to the respective server regardless of the network distance. After the users are clustered into groups and assigned to a server, the specific location of the server is recalculated to minimize the maximum distance to every assigned user. In comparison to Experiment 1, we obtain a local optima for the minimized cache miss ratio for every specified amount of servers. Figure 6 shows the connection between profile correlation and cache miss ratio as well as the impact of different amount of servers. A server caches up to 10% and a user requests 15% of the services in the universe. We illustrate the gap between optimized distance and optimized profile correlation.

The curves show the potential benefits for the CDN provider or ISP depending on the optimization criteria. It highlights the advantages of a cooperation for a common system approach. The optimization

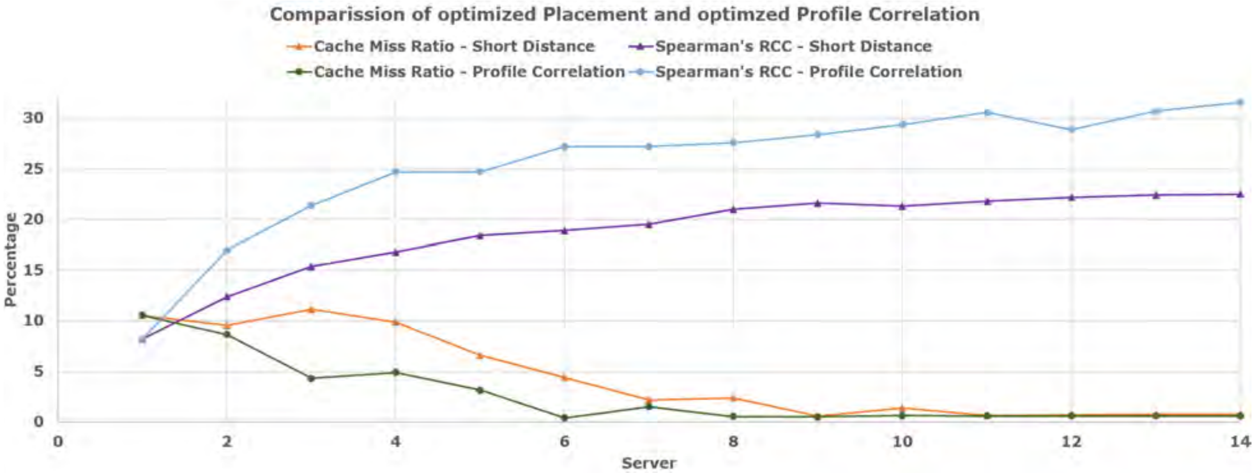

Figure 6: Optimization of high profile correlation and simulation of the cache miss ratio for different amount of server using the example scenario in Section 2.

towards profile correlation leads to an improved cache hit ratio. With an increased amount of servers, the profile correlation coefficient stagnates or gets worse. This is due to the performance of the profile optimization algorithms and the complexity of the problem. Nevertheless, the impact on the cache is minimal due to the already improved caching behavior and high amount of servers. For small amount of servers, the caching behavior is improved heavily. In some situations, the cache miss ratio can be reduced to half and even more. Additionally, an improved profile correlation does not automatically lead to an improved caching behavior in simulation, for example shown in the case from 3 to 4 servers due to cache size. The average distance from a user to the server is higher and varies more if we optimize according to profile correlation instead of distances. These results support the provider in decisions for additional servers to improve the system behavior and service level to users.

Experiment 3: In this experiment, we analyzed the effective size of the cache and evaluate an adjusted replacement algorithm. The scenario uses 5 servers, their location and user assignments are fixed. Every server has the same amount of cache and the user profile contains 15 services. Figure 7 shows the cache miss rate for different sizes of cache. It compares the performance of several cache replacement algorithms.

With increased size of cache, the improvement is reduced with every additional cache space until it stagnates. LRU-2, LFU and LIRS are close to the global optimum calculated in accordance with Belady. With a cache size of 12, we reach the minimum, whereas increased cache size has negligible impact on the cache miss ratio. Even if the spectrum of a user profile has more entries, the additional services are not requested very often due to Zipf-Distribution. So it has negligible impact on the general caching behavior.

Experiment 4: In line with our initial intention, we calculate a local Pareto Front with our approach for average network distance and profile correlation from a user to its primary server. The Figure 8 shows the effect of varying server numbers and their impact on the caching behavior as well as the network load.

The AMOSA algorithm calculated only 4 solutions at the Pareto Front. In contrast, our approach reached more fine grained solutions. The comparison for 4 servers shows that our combined optimization algorithms calculate a better Pareto Front. With increasing amount of server, the average length of the network path is reduced. The “optimal” balance between distance and profile correlation depends on the goals of the CDN provider and ISP. It depends on agreements between the partners and respective calculations can be made for every point of the Pareto Front. The Pareto front provides a set of candidate solutions for the negotiations between CDN provider and ISP.

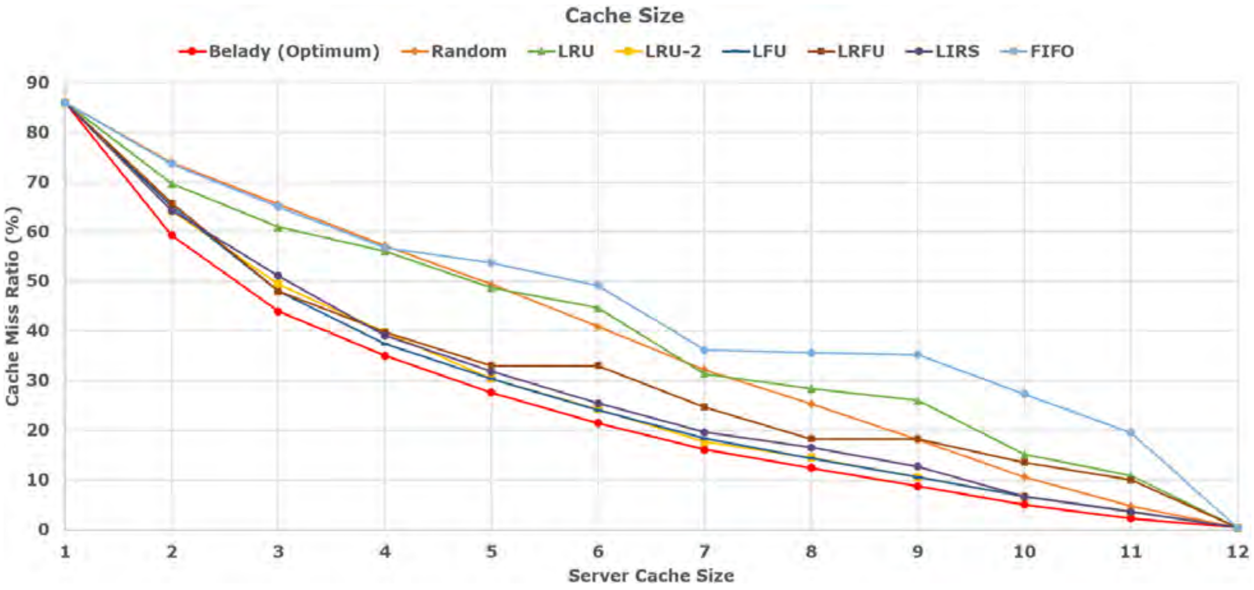

Figure 7: Cache miss ratio for different sizes of cache.

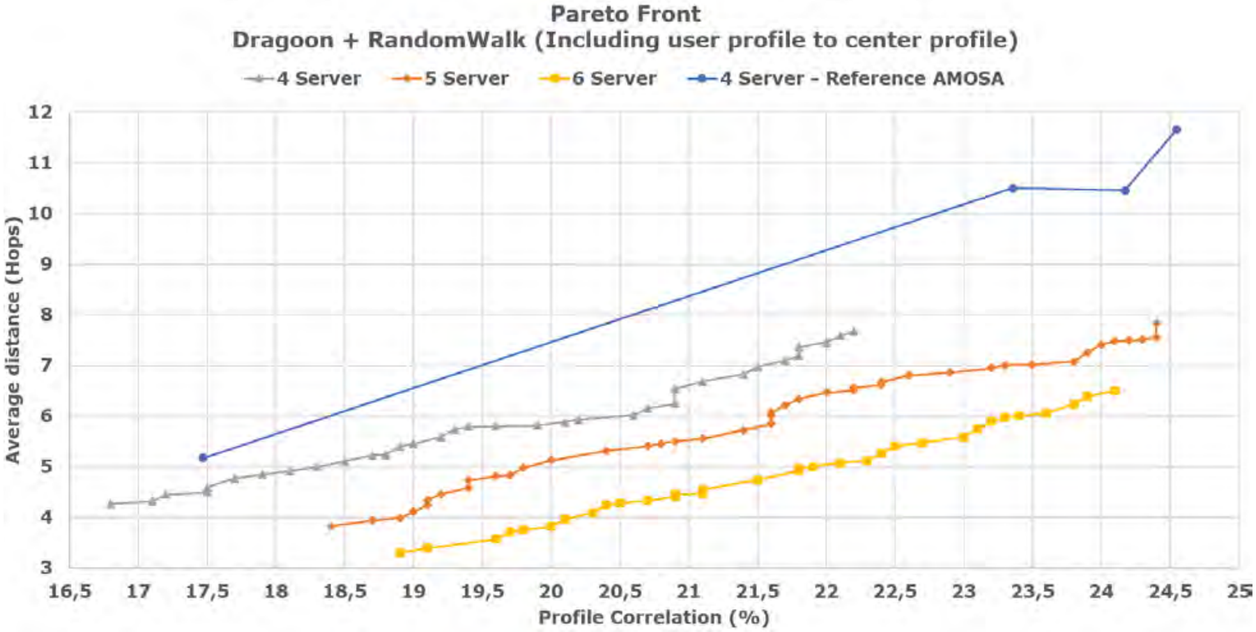

Figure 8: Solutions at the local Pareto Front with algorithms Dragoon as initialization with optimized placement and Random Walk for iterative improvement towards profile correlation.

7 CONCLUSION AND OUTLOOK

In this paper, we propose a model to simulate and optimize a CDN according to multiple objectives. We model different user behaviors and take advantage of their profiles during optimization. Our described model considers multiple challenges in the storage chain, which are interacting with each other. We propose a novel approach to improve the entire processes. This enables an overall optimization to find combined solutions and highlights the cooperation between a CDN provider and ISP to reach a high service level. With this model, we are able to simulate and analyze overall system processes and behaviors under specific constrains. We provide a tool to support the management in its decisions process with regard to the necessary amount of servers, the placement of these servers and the recommended cache size. Additionally, we improve the assignment of users to servers with fitting content. We simulated different system behaviors and visualized special aspects during runtime. In the future, we plan to include further organization strategies for the servers to balance the content and load.

ACKNOWLEDGMENTS

This work was partly funded by FLAMINGO, a Network of Excellence project (ICT-318488) supported by the European Commission under its Seventh Framework program.

REFERENCES

Alimi, R., R. Penno, Y. Yang, S. Kiesel, S. Previdi, W. Roome, S. Shalunov, and R. Woundy. 2014, September. “Application-Layer Traffic Optimization (ALTO) Protocol Request for Comments: 7285”. Standards Track ISSN: 2070-1721, Internet Engineering Task Force (IETF).

Bandyopadhyay, S., S. Saha, U. Maulik, and K. Deb. 2008, June. “A Simulated Annealing-Based Multiobjective Optimization Algorithm: AMOSA”. In *Proceedings of the Transactions on Evolutionary Computation*, Volume 12, 269–283. Piscataway, New Jersey: Institute of Electrical and Electronics Engineers, Inc.

Belady, L. A. 1966, June. “A Study of Replacement Algorithms for a Virtual-storage Computer”. *IBM Systems Journal* 5 (2): 78–101.

Breslau, L., P. Cao, L. Fan, G. Phillips, and S. Shenker. 1999, March. “Web Caching and Zipf-like Distributions: Evidence and Implications”. In *Eighteenth Annual Joint Conference of the IEEE Computer and Communications Societies (INFOCOM).*

Frank, B., I. Poese, Y. Lin, G. Smaragdakis, A. Feldmann, B. Maggs, J. Rake, S. Uhlig, and R. Weber. 2013, July. “Pushing CDN-ISP Collaboration to the Limit”. *SIGCOMM Computer Communication Review* 43 (3): 34–44.

headissue GmbH 2016. “cache2k: Benchmarks”. Accessed July 7, 2016. http://cache2k.org/benchmarks.html.

Jamin, S., C. Jin, A. R. Kurc, D. Raz, and Y. Shavitt. 2001, April. “Constrained Mirror Placement on the Internet”. In *Twentieth Annual Joint Conference of the IEEE Computer and Communications Societies (INFOCOM)*, Volume 1, 31–40. Anchorage, AK.

Jiang, S., and X. Zhang. 2002. “LIRS: An Efficient Low Inter-reference Recency Set Replacement Policy to Improve Buffer Cache Performance”. In *Proceedings of the 2002 ACM SIGMETRICS International Conference on Measurement and Modeling of Computer Systems*, SIGMETRICS ’02, 31–42. New York, NY, USA: ACM.

Jiang, W., R. Zhang-Shen, J. Rexford, and M. Chiang. 2009. “Cooperative Content Distribution and Traffic Engineering in an ISP Network”. In *Proceedings of the Eleventh International Joint Conference on Measurement and Modeling of Computer Systems*, SIGMETRICS ’09, 239–250. New York, NY, USA: ACM.

Kleinberg, J., and E. Tardos. 2013, December. *Algorithms Design*. Pearson-Addison Wesley. Lecture slides by Kevin Wayne.

Knight, S., H. Nguyen, N. Falkner, R. Bowden, and M. Roughan. 2011, October. “The Internet Topology Zoo”. *Selected Areas in Communications*.

Krishnan, R., H. V. Madhyastha, S. Srinivasan, S. Jain, A. Krishnamurthy, T. Anderson, and J. Gao. 2009. “Moving Beyond End-to-end Path Information to Optimize CDN Performance”. In *Proceedings of the 9th ACM SIGCOMM Conference on Internet Measurement Conference*, IMC ’09, 190–201. New York, NY, USA: ACM.

Lee, D., J. Choi, J. H. Kim, S. H. Noh, S. L. Min, Y. Cho, and C. S. Kim. 2001, December. “LRFU: A Spectrum of Policies That Subsumes the Least Recently Used and Least Frequently Used Policies”. *IEEE Transactions on Computers* 50 (12): 1352–1361.

Lloyd, S. 2006, September. “Least Squares Quantization in PCM”. *IEEE Transactions on Information Theory* 28 (2): 129–137.

Macqueen, J. 1967. “Some Methods for Classification and Analysis of Multivariate Observations”. In *5-th Berkeley Symposium on Mathematical Statistics and Probability*, 281–297.

Mobasher, B., R. Cooley, and J. Srivastava. 2000, August. “Automatic Personalization Based on Web Usage Mining”. *Communications of the ACM* 43 (8): 142–151.

Pallis, G., and A. Vakali. 2006, January. “Insight and Perspectives for Content Delivery Networks”. *Communications of the ACM* 49 (1): 101–106.

Qiu, L., V. N. Padmanabhan, and G. M. Voelker. 2001, April. “On the Placement of Web Server Replicas”. In *Proceedings of IEEE Infocom Conference*. Anchorage, AK.

AUTHOR BIOGRAPHIES

PETER HILLMANN is a Ph.D. student at the Universität der Bundeswehr München (UniBwM), Germany. He holds a M.Sc. in Information-System-Technology from Dresden University of Technology since 2011. His areas of research are network and system security with focus on cryptography as well as operational modeling and optimization. His email address is peter.hillmann@unibw.de.

TOBIAS UHLIG is a postdoctoral researcher at the Universität der Bundeswehr München, Germany. He holds a M.Sc. degree in Computer Science from Dresden University of Technology and a Ph.D. degree in Computer Science from the Universität der Bundeswehr München. His research interests include operational modeling, natural computing and simulation-based optimization. He is a member of the ASIM and the IEEE RAS Technical Committee on Semiconductor Manufacturing Automation. His email address is tobias.uhlig@unibw.de.

GABI DREO RODOSEK is Professor of Communication System and Network Security at the Universität der Bundeswehr München, Germany. She holds a M.Sc. from the University of Maribor and a Ph.D. from the Ludwig-Maximilians University Munich. She is spokesperson of the Research Center Cyber Defence (CODE), which combines skills and activities of various institutes at the university, external organizations and the IT security industry (for instance Cassidian, IABG or Giesecke & Devrient). Her email address is gabi.dreo@unibw.de.

OLIVER ROSE holds the Chair for Modeling and Simulation at the Department of Computer Science of the Universität der Bundeswehr München, Germany. He received a M.S. degree in applied mathematics (1992) and a Ph.D. degree in computer science (1997) from Würzburg University, Germany. His research focuses on the operational modeling, analysis and material flow control of complex manufacturing facilities, in particular, semiconductor factories and assembly systems. He is a member of INFORMS Simulation Society, ASIM (German Simulation Society), and GI (German Computer Science Society). In 2012, he served as General Chair of the WSC. Currently, he is member of the board of the ASIM and the ASIM representative at the Board of Directors of the WSC. His email address is oliver.rose@unibw.de.