Optimizing Server Storage Capacity on Content Distribution Networks

Felipe Uderman[1], Tiago Neves[1], Célio Albuquerque[1]

[1]Instituto de Computação – Universidade Federal Fluminense (UFF)
Niteroi – RJ – Brazil

{uderman,tneves,celio}@ic.uff.br

Abstract. *This work addresses the Storage Capacity Allocation Problem (SCAP), which is closely related to Content Distribution Networks (CDNs) planning and management. The problem consists of determining optimally the fraction of the total storage that must be allocated to each CDN server. A survey of the relevant bibliography on the subject is provided. In this work we have designed and implemented an exact method in order to solve the SCAP problem, and have also evaluated the performance of a CDN before and after the storage capacity optimization. Our simulation results show that an optimally storage capacity configuration has a major impact on the costs related to the delivery of contents.*

1 Introduction

A Content Distribution Network (CDN) is an overlay network of collaborative servers where content replicas are placed and then delivered to its clients [Rajkumar Buyya (2008)]. Because of the increasing amount on the number of requests for contents on the Internet, CDNs have become a popular choice for content providers that want to reach its current and potential clients. This statement is valid especially for multimedia contents, which usually have more severe Quality of Service (QoS) constraints. A CDN is able to place servers, frequently called cache servers or surrogate servers, near its clients, to replicate contents on them and also to process clients' requests for contents, designating which server will handle each request. These functionalities imply on an improvement of content availability and, therefore, result on positives effects on the performance of the service from the perspective of clients, that will experience a reduction on network delay and an improvement on reception rate in most cases.

The current Internet architecture can not enforce acceptable levels of QoS to its users, as it operates on a best effort model. Despite the existence of protocols that are able to provide QoS guarantees on TCP/IP networks, the implementation of such protocols in global scale can be very challenging, due mostly to the distributed management of the Internet and to the ossification of its protocols and services. Still, there is a great demand for improvements on the QoS levels provided by the Internet that motivates the development of alternative solutions to accomplish this objective. Therefore, the main purpose of a CDN is to mitigate current Internet deficiencies and undesirable characteristics, in order to deliver contents to its users with an acceptable QoS. To accomplish this challenging task while still being able to profit, CDN providers must apply optimized mechanisms that aim to reduce its implementation and operational costs.

One important portion of a CDN dimensioning is to determine the storage capacity of its servers. The employment of an over dimensioned storage capacity on CDN servers may contribute to an improvement on CDN performance. In an over dimensioned scenario, servers are able to hold more contents and therefore, they may handle requests of nearby clients more easily. However, an over dimensioned server storage is obviously not desirable due to the costs involved and to the high probability of under usage of the storage capacity. On the other hand, a sub dimensioned storage capacity will probably lead to lower average performance, since in such condition CDN servers can not hold enough contents to handle most clients' requests with an acceptable performance. In case a content requested by a client is not present on a nearby server, the client will have to wait until either the CDN forwards its request to another server, or until the contacted server retrieves the desired content from another server. As the CDN providers have a limited budget to employ in their physical structure, it is of major importance to develop efficient techniques to aid CDN providers to efficiently dimension the storage capacity of each server in order to achieve an acceptable level of QoS imposed by a set of geographically distributed clients. Therefore, the problem addressed in this paper, known as Storage Capacity Allocation Problem (SCAP), is very relevant for CDN providers that aim to employ efficient infrastructures of collaborative servers.

On this paper, we implement and evaluate a solution for SCAP by modeling it as a Integer Linear Programming Problem (ILPP). The ILPP implemented can obtain the optimal solution for the SCAP on CDNs where any server is able to handle requests from any client. Therefore, upon a request for a content, the server that can perform the delivery more efficiently will be utilised whenever possible. If this particular server does not hold the requested content, the request will be routed to another CDN server that has the desired content and the missing content will be copied to the originally chosen server. It is important to highlight that the ILPP implemented for this particular CDN model can easily be modified to suit other models. We evaluate the effectiveness of optimizing the storage capacity of a CDN, by comparing the performance of test instances with the storage capacity of the servers given by an uniform distribution against instances with optimized storage capacity allocation. Our simulation results demonstrate that a CDN with an optimized storage capacity allocation performs significantly better when compared to CDNs with non optimized storage capacity allocation.

The remaining of this paper is organised as follows: In Section 2 we analyse the SCAP, focusing on the necessary input data to solve this problem and also on possible variations regarding its mathematical modeling and expected results. In Section 3 we review relevant related works available on the literature. In Section 4 a mathematical model to solve the SCAP is presented and analysed. In Section 5 we detail the test scenarios, and also introduce the mechanism used to evaluate the effectiveness of the storage capacity allocation on CDNs. In Section 6 we present and analyse our simulation results. We conclude our work on Section 7, by consolidating the results obtained and proposing possible future works.

2 The Storage Capacity Allocation Problem (SCAP)

Many of the challenges faced by CDN providers can be modeled as optimization problems, whose solutions can be used to assist on the dimensioning of CDN resources and on the development of efficient algorithms and strategies to perform content replication and delivery. There are three fundamentals optimization problems related to CDNs dimensioning and operation:

- **Server Location Problem (SLP):** This problem consists on determining the optimal location of the available cache servers in order to better server the clients [Krishnan (2000)].
- **Request Distribution Problem (RDP):** This problem consists on determining optimally which server must be responsible for delivering the content to each client [Wang (2002), Shaikh (2001)].
- **Replica Placement Problem (RPP):** Due to capacity constraints, the cache servers available don't have enough storage capacity to hold all the contents of the CDN system thus, the RPP consists on determining optimally which contents must be stored at each server on each time period, and also on determining the servers that will be used as seeds during the replication process [Khan (2009)].

The problem stated on this work, known as the Storage Capacity Allocation Problem (SCAP), is a variant of the SLP. The SCAP consists on optimally determining the distribution of the total storage capacity available among the surrogate servers of the CDN. Solutions for the SCAP require input data that describes characteristics and limitations of the CDN, such as the total storage capacity available, the communication costs between each element of the CDN and the CDN topology and operational model. Besides that, information about the clients of the CDN are required in order to determine the relative popularity of each content among the clients and also how many requests for contents each client will perform over the time periods analysed.

Besides determining the storage capacity allocation for each server, some versions of the SCAP can also determine, in a static manner, which server should handle each request and also which contents should be placed on each server, characteristics related to the RDP and RPP, respectively. However, as dynamic approaches for the RDP and the RPP have been proved to be effective [Neves (2010)], these static additional results provided by the SCAP should only be utilised to determine the initial location of content replicas and request distribution, leaving to dynamic mechanisms the solution of the RDP and RPP during the CDN operation phase.

Although there is a decreasing tendency on the costs of storage for servers, the SCAP can become a major issue on future scenarios, because of the rapid growth of the demand for content on the Internet, especially for multimedia content. Some kinds of multimedia content, like high resolution videos, which still have a low availability on current Internet, have a considerable size when compared to ordinary contents, and can easily exhaust the storage capacity of CDN servers if they become popular and widespread. The small availability of high speed connections on the last mile of the Internet is still the main cause of the low popularity of services specialised in delivering high resolution multimedia content. But on scenarios where the majority of the end users have access to high speed connections, the bottleneck of those services will become the availability of contents on the servers. As centralized approaches to content delivery has severe scalability problems and P2P networks can not ensure QoS guarantees, we reinforce the importance of the SCAP on the planning of CDN infrastructures that aim to support an efficient delivery of widespread high storage and bandwidth demanding contents over the Internet.

An important issue of the SCAP is that it is essentially an offline problem, meaning that all the information required to solve it must be a priori available. Thus, some input parameters of the SCAP that have a dynamic behaviour, such as the communication costs between the servers, magnitude of contents demands and number of clients, must be either accurately estimated or extracted from some data set that describes a CDN operation scenario. This work follows the second approach, by extracting the required information from test instances for the Replica Placement and Request Distribution Problem (RPRDP), a joint problem of the RDP and the RPP, which deals with many issues related to the operation of a CDN simultaneously. Thus, as the mathematical modeling of the SCAP benefits from this kind of information, the results obtained must be considered as theoretical bounds of performance, as in practice, the information required to compute the input parameters for the SCAP can not be exactly obtained.

Another limitation is that traditional solutions for the SCAP will distribute the available storage capacity in a very granular fashion [Laoutaris (2005)], ranging from a single store unit to content size increments. This means that, even with all the input parameters at hand, the SCAP will generate an optimal storage capacity distribution that can not be implemented in practice, as real devices storage capacity is available in few discrete sizes, usually with a considerable amount of storage units apart from each other. Even so, the optimal results generated by SCAP can easily be used to aid the choice of the best available server capacity for each location.

A realistic scenario, where is possible to use the SCAP results accurately, is the one where the CDN servers are implemented as virtual machines that shares computational resources with other applications and services. On this scenario, due to virtualization of the servers, it is easier to allocate storage capacity for each CDN content server with a high granularity. In addition, it would be possible to repeatedly use the static model of the SCAP to solve the dynamic version of the problem, on which the storage capacity of each virtual server could be adapted to provide the most suitable storage capacity distribution at each time period. Even on this dynamic scenario, the SCAP could be modeled as an offline ILPP that would provide an optimal solution based on the information available for all the time periods, providing a theoretical bound on the performance for this dynamic version of the problem. However, the most reasonable approach for this dynamic version is to implement an efficient method to solve the problem on an online fashion, where the storage capacity distribution must be determined, for each time period, based solely on information of the current and previous time periods.

3 Related Works

In this section, previous publications that have guided the development of this work are analysed. On [Laoutaris (2005)], the authors present the SCAP for CDNs with hierarchical topologies. Three mathematical models are proposed for different versions of the SCAP, along with greedy heuristics that have archived results close to optimum on the simulations performed. Due to the high computational complexity of the models, the LP relaxed version of the models are solved instead of the original ILPPs. The first model, suited for CDNs with a strong hierarchical relationship between its servers, is the simplest one as it does not consider additional features and limitations that could improve the model verisimilitude. The following models include a load balance constraint, that limits the maximum number of requests per unit time that may be serviced by each server, and a request peering capability, that allows CDN servers to redirect requests to other servers that belong to its same hierarchical level. The addition of the load balance constraint increases the total cost of the solution, but causes the usage of the different CDN servers be more uniform. The peer requesting feature reduces significantly the total cost of the solution, specially when the communication cost between peer servers is low.

On [Li (1999)], the authors address the SLP as a dynamic programming problem, with the objective of determining the optimum location of web proxy servers on the Internet while minimising the delay experienced by clients when locating and accessing contents on the CDN. Although the SLP can solve the problem of the location of the surrogate servers on a CDN, no results can be obtaining on the dimensioning of each server storage capacity, which represents a drawback of this model. Besides, the presented scenario supposes that all network links have a homogeneous capacity and the existence of web proxy servers for a CDN with only one original server, characteristics that do not match with realistic scenarios.

On [Qiu (2001)], the authors have proposed several heuristic algorithms to solve the SLP, and have evaluated their performance over synthetic and real network topologies derived from BGP routing tables. Among the heuristic algorithms proposed, a greedy Algorithm that aims to individually determine which location is better suited to place a cache server on each interaction has archived the best results. The authors also comment about practical issues of obtaining the input data necessary to solve the SLP on realistic scenarios. Although it is relatively simple to obtain accurate information about the network topology being analysed, it is not so easy to obtain information of some parameters accurately, such as the performance of the communication links and the amount of clients requests for contents, as those parameters have a very dynamic nature. Nevertheless, simulation results show that the placement algorithms proposed are relatively insensitive to errors on the estimate of those parameters.

On [Wu (2009)], the authors also address the SLP, but their solution is based on a genetic algorithm approach. A numeric solution for the same network topology analysed on [Li (1998)] is presented, along with a performance comparison of the genetic algorithm approach with the greedy algorithm proposed on [Qiu (2001)]. The greedy algorithm was chosen for performance evaluation because it outperforms the other algorithm proposed on [Qiu (2001)] and the dynamic programming algorithm proposed on [Li (1998)]. The numeric results demonstrate that, for a number of available servers greater than two, the genetic algorithm can achieve better results then the greedy algorithm. However, like other similar works on this issue, no consideration about the dimensioning of the servers capacity is made. Beside, as the numeric results were computed only for one network topology with homogeneous network links capacity, it is not possible to conclude that the genetic algorithm proposed will have a good performance for other network topologies. Indeed, the authors shows that the performance gain of the genetic algorithm proposed, when compared to the greed solution of [Qiu (2001)], can vary significantly when different sets of request rates are assigned to clients.

On [Li (2008)], the authors address the RPP on the context of hierarchical topologies. The optimal solution for the formulated RPP is obtained by solving a dynamic problem. The simulations performed compare the performance of several replica positioning protocols on scenarios with a variable total storage available and also with topologies with variable hierarchical levels. The results suggest that a correct dimensioning of the total storage capacity on a CDN is an important parameter for performance. It is also possible to conclude that the number of hierarchical levels of a CDN should not be superior to four because, despite the moderate increase of the cache hit ratio on hierarchical CDN topologies with more levels, there is a huge increase on the average delay observed by the clients due to the increase on the difficulty for locating contents on servers. Furthermore, simulations performed with different values of the parameter of the distribution used to model the content popularity on the CDN show that scenarios in which clients' demands are concentrated on few contents are advantageous to the proposed solution.

By analysing the referred literature, it is possible to conclude that related works that addressed storage allocation issues on CDNs [Laoutaris (2005), Li (1998), Wu (2009)] have not extrapolated their results by evaluating the performance of a CDN operation when its total storage capacity is optimally distributed or when the network nodes that hold surrogate servers are optimally chosen. Indeed, those works limit the scope of their analysis to the performance of the SCAP or SLP solutions proposed, by analysing the computational complexity of the solution or by proposing efficient heuristics algorithms to solve those problems. We believe that, by evaluating the impact of allocating optimally the storage capacity of surrogate server on the performance of a CDN operation, we can provide a contribution to the scientific community and to CDN providers as well.

4 Mathematical Modeling

The SCAP can be formally defined as follows [Laoutaris (2005)]: let S be the total storage capacity available, φ the set of N unique unit sized contents, J the set of m clients, each client j having a request rate λ_j and a object demand distribution $p_j(k) \rightarrow [0,1]$, V the set of n servers, $d_{j,v} : J \times V \rightarrow R^+$ the communication cost between the clients j and the node v, and C the set of all the possible node-objects pairs $(v,k), v \in V, k \in \varphi$. The SCAP consists on determining a subset $A \subset C$ with no more then S elements that maximize the gain obtained when a client of the CDN receives a content from a surrogate server instead of receiving it directly from the content server.

The mathematical modeling of SCAP must consider parameters related to the RPRDP to ensure that the storage capacity allocation is optimal to a given CDN model. For example, on a CDN operational model that bounds each client to a local server, it makes no sense to consider the possibility that other CDN servers might be able to deliver contents to then. Similarly, on a CDN with hierarchical topology, only a subset of servers are able to deliver contents to each client. The CDN model utilised to evaluate our results operates with a full-mesh topology, where every server of the CDN is able to deliver contents to a given client.

The SCAP can be modeled as ILPP in order to compute the optimal solution of the problem. Firstly, it is necessary to define two variables:

$$X_{j,v}(k) = \begin{cases} 1 & \text{, if the client } j \text{ receives the} \\ & \text{content } k \text{ from the server } v \\ 0 & \text{, otherwise} \end{cases} \tag{1}$$

$$\delta_v(k) = \begin{cases} 1 & \text{, if } \sum_{j \in J} X_{j,v}(k) > 0 \\ 0 & \text{, otherwise} \end{cases} \tag{2}$$

The variable $X_{j,v}(k)$, defined on Equation 1, implies on whether the content k will be delivery by the server v to the client j and the variable $\delta_v(k)$, defined on Equation 2, implies on whether the content k will be present on the server j. By computing $\delta_v(k)$, it is possible to obtain the storage capacity allocated on each node of the CDN. The ILPP adapted for solving the SCAP for full-mesh CDN topologies is defined as:

Max:

$$\sum_{j \in J} \lambda_j \sum_{k \in \varphi} p_j(k) \sum_{v \in V} (d_{j,os(k)} - d_{j,v}) X_{j,v}(k) \tag{3}$$

S.A.:

$$\sum_{v \in V} X_{j,v}(k) \leq 1 \tag{4}$$

$$\sum_{j \in J} X_{j,v}(k) \leq U \cdot \delta_v(k) \tag{5}$$

$$\sum_{v \in V} \sum_{k \in \varphi} \delta_v(k) \leq S \tag{6}$$

$$\sum_{k \in \varphi} \delta_v(k) \geq |\{k : OS(k) = v\}| \tag{7}$$

The Equation 3 is the objective function of the ILPP and consists on maximizing the gain obtained when storage space for content k is allocated on server v. The gain is proportional to the difference of the network distances from client j to the content original server and from client j to server v. This means that more storage capacity will be allocated on servers that have a high potential of reducing the network load or improving the QoS perceived by the users, depending of the distance metrics applied. The next term of Equation 3 is relative to the request rate of client j. The request rate should represent the amount of requests originated on of client j. The number of users on a client site can be utilised to derive the request rate, but it is likely that different client sites will have users with different activity profiles. Finally, the request distribution component takes into consideration that each content will have a different popularity on different client sites.

Equations 4, 5 and 6 are the constrains that must be observed in order to keep the consistency of the solution. Equation 4 states that a client j must receive each content k from a single server v. This assumption is necessary to ensure that the storage capacity allocation will consider the best server to address each request, but it is the responsibility of the RPRDP method implemented to dynamically make this decision. Equation 5 states that a client j can only receive a content k from a server v if this server have storage capacity allocated for this content, and also that the number of clients that receives content k from server j must not exceed the total number of clients of the CDN. Equation 6 states that the amount of storage capacity allocated must not exceed the total storage capacity available. Equation 7 states that the amount of space allocated on a given server must be enough to place at least the contents that originate on that CDN server. This avoids inconsistencies while processing the RPRDP, as at its initial period, there must be enough storage space on each server to accommodate the contents.

Table 1. Relevant RPRDP parameters

Parameter	Description
Content size	Uniformed distributed between 250MB and 400MB
Content origin Server	Random
Server capacity	Uniformed distributed between 3000MB and 4000MB
Contents popularity	Zipf distributed, $\alpha = 0.7$[Adamic (2002)]
Requests local server	Zipf distributed, $\alpha = 0.7$
Simulation periods	25
Request arrival time periods	Time periods 0 to 14

5 Test Scenarios

The simulations performed aim at identifing improvements on the performance of a CDN with an optimised storage capacity allocation, in contrast with a CDN with an uniformly distributed storage capacity allocation. To accomplish this task, we have generated test instances for the SCAP by extracting the necessary information from test instances for the RPRDP [Neves (2010)]. With the SCAP results, we have generated new test instances for the RPRDP with an optimized storage capacity allocation, keeping all the other parameters unchanged. This means that differences in performance observed between the original and optimised instances of the RPRDP must be credited solely to the new storage scheme.

The RPRDP is a typical operation scenario of a CDN, where it is necessary to dynamically define which contents will be stored on each servers and also which server will attend each client request for contents. Therefore, it is possible to extract from these instances the necessary information necessary to model the SCAP, since the RPRDP instances describe the requests for contents events, which can be used to obtain $p_j(k)$ and λ_j, besides other necessary parameters such as total storage capacity available, amount of contents and communication costs between the clients and the servers. We have created a total of 140 instances for the RPRDP, segmented in groups of 10, 20, 30 and 50 servers. Some of the parameters of the original RPRDP instances that are relevant for the SCAP are described on Table 1.

The objective function of the RPRDP computes the costs associated with the delivery of contents to the CDN clients and with the replication of contents among the CDN servers, as it can be observed on Equation 8. The delivery cost is associated with the network delay between a client and a server, at the time period a content is being delivered. This parameter models the user perceived QoS, as a lower network delay will result on a faster and more reliable delivery. The replication cost is associated with how much data is being replicated among the CDN servers. This parameter models the CDN provider costs with their communication infrastructure. If less content needs to be replicated, it means that the CDN providers will spend less resources on their communication infrastructure. There is also a backlog term on the RPRDP objective function, that models situations where a requested content can not be delivered to a client upon a request. Therefore, the backlog represents the amount of data that could not be delivered to the client, being essentially a measurement of the client satisfaction to the service provided by the CDN. We have verified that the backlog is caused by lack of bandwidth with our RPRDP implementation. As our test instances have enough bandwidth to attend the clients requests, no backlog cost was observed on our simulations. The RPRDP minimizes the following cost function:

$$Cost = \sum_{i \in R} \sum_{j \in S} \sum_{t \in T} c_{ijt} x_{ijt} + \sum_{i \in R} \sum_{t \in T} p_{it} b_{it}$$

$$+ \sum_{k \in C} \sum_{j \in S} \sum_{l \in S} \sum_{t \in T} L_k w_{kjlt} \qquad (8)$$

Where:

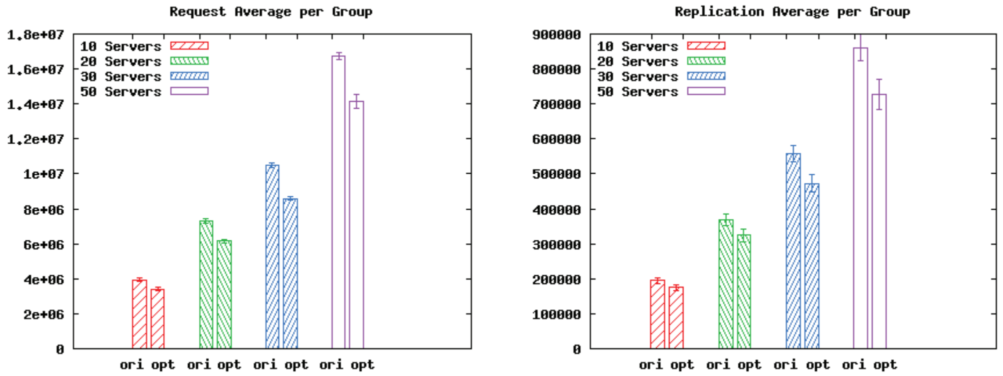
- c_{ijt}: Cost of delivery of the content k requested on i by the server j, on the time period t
- x_{ijt}: Fraction of the content requested on i delivered by server j, on the time period t
- p_{it}: Backlog penalty of request i on time period t
- b_{it}: Backlog amount of request i on time period t
- L_k: Size of content k
- w_{kjlt}: 1, if content k is replicated from server l to server k on the time period t; 0, otherwise
- $i \in R$: Set of requests
- $j, l \in S$: Set of servers
- $t \in T$: Set of time periods
- $k \in C$: Set of contents

6 Simulation Results

In this section the simulation results of evaluating the performance gain of a CDN with an optimally distributed storage allocation in contrast with a CDN with an uniform distributed storage allocation are presented. The results of the 35 instances of each group of 10, 20, 30 and 50 servers were summarised for better visualisation.

Figure 1 shows costs averages for each group of instances along with the 95% confidence interval. For all groups of instances, the optimization of the storage allocation have reduced significantly both the replication and the delivery costs of the RPRDP. This means that both the CDN provider and its clients can benefit from an optimized storage capacity allocation, even if this optimization is performed statically. As the request cost is related to the network distance between the client and the CDN server assigned for content delivery, a lower request cost average means that the client would experience a faster and more reliable delivery. The replication cost is related with how much data must transfer among the CDN servers, and it is related to the costs of the internal infrastructure of the CDN. Therefore, a lower replication cost represents savings for the CDN provider, as less resources would be spent on its infrastructure.

Another improvement achieved is the increase of the cache hit ratio on the local CDN server, as can be observed on Figure . For each CDN client, it is assigned a local server that is closer to it and therefore must be the first choice for content delivery. An improved cache hit ratio on the local server also benefits both the client experience and the CDN provider costs, as delivering a content from a remote server will deteriorate the user



(a) Request (b) Replication

Figure 1. Cost comparison between optimized and uniform distributed storages allocation

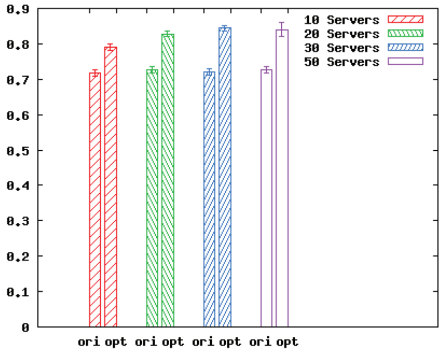
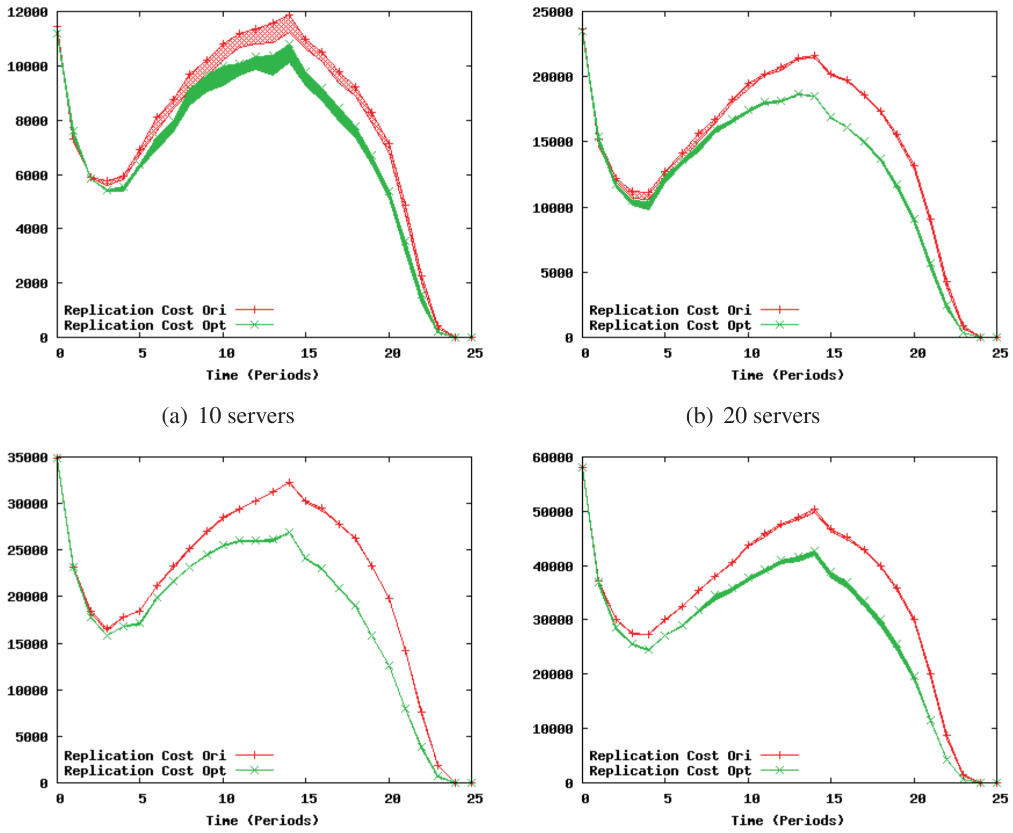


Figure 2. Cache Hit Ratio on the local server

perceived quality of delivery and also will represent more data transit on the CDN internal infrastructure.

Figure 3 shows the replication costs for each group of instances summarised by each time period of simulation. Both the original and optimised instances follows the same behaviour, but the optimized instances perform about 10% to 20% less replication over almost all the time periods of simulation. At the very begin of simulation, it is necessary to perform a high amount of replication, as the CDN servers are not yet populated with contents. Therefore, the replication cost is high at the very begin of the simulation and keeps dropping until around period 4. Beyond this period, as the number of active requests on the CDN keeps rising, more replication is performed in order to better address the requests. This rise is observed until the end of request generation on time period 14, where it is possible to observe the higher amount of replication performed. After the end of new requests generation, the number of active requests keeps droping, and less replication is necessary until the last request is finally completed around time period 24.

On Figure 4, it is possible to observe the delivery cost for each time period of simulation. At the begining of the simulation, the contents have not yet been replicated among the CDN servers. Therefore, the content delivery is performed from remote servers, increasing the delivery cost. After content replication occurs, the delivery cost drops for a few time periods, but the accumulation of active requests makes the delivery cost rise until around time period 14, when the request generation ends. After this time period, the



(a) 10 servers

(b) 20 servers

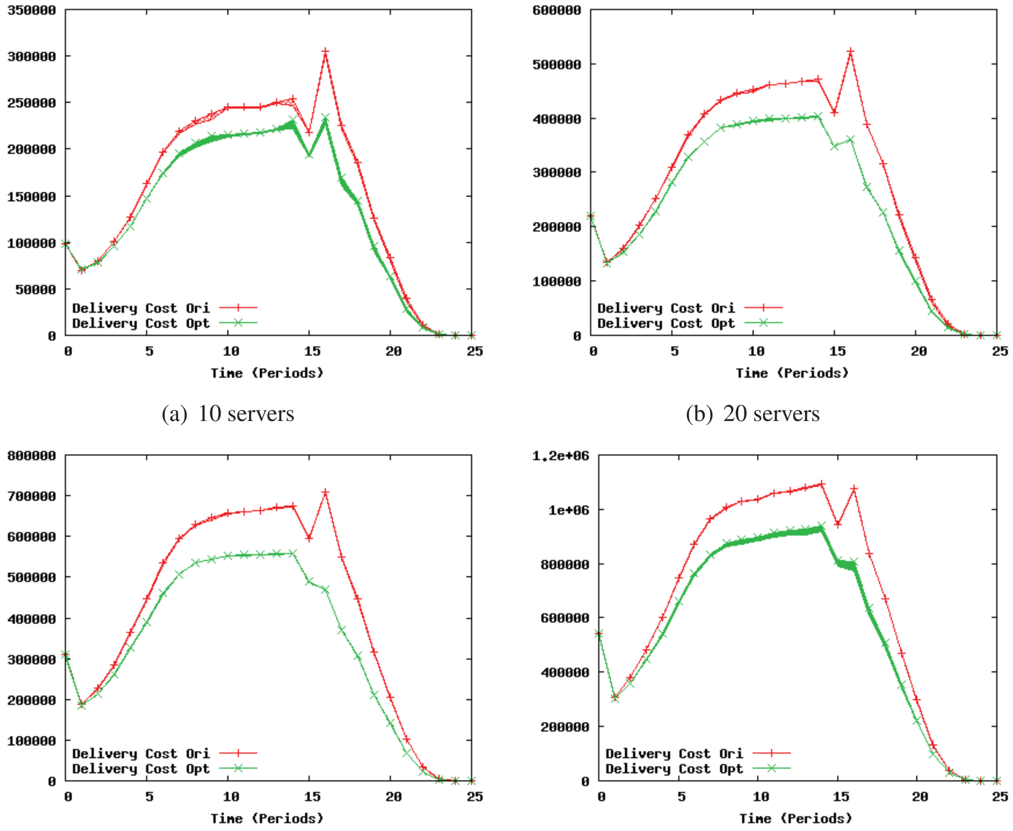
(c) 30 servers

(d) 50 servers

Figure 3. Replication Costs over time comparison

delivery request cost starts to drop, but around time period 17 there is a sudden increase on the delivery cost for the non optimized instances. On the optimized instances, even when this spike is not observed on all of the optimized instances groups, it is still possible to observe an attenuation on the dropping rate. As can be observed on Figure 5, this behaviour can be explained by a drop on the cache hit ratio. Less requests are addressed by the local servers, what impacts the delivery costs even with less active requests present on the CDN.

Figure 5 presents the cache hit ratio on the local server for each time period of simulation. At the initial time period of simulation, the cache hit ratio is very poor for both optimized and non-optimized instances, because the contents have not yet been replicated among the CDN servers. After replication occurs, the cache hit ratio improves fast, but as the number of active requests on the CDN rises, it is not possible to maintain the cache hit ratio on the local server at a high level. It is also possible to observe a drastic drop on the cache hit ratio around time period 17. This behaviour can be explained by the simplicity of the replication heuristic utilised to evaluate the storage capacity optimization performed. We have observed that, as the removal content policy of the replication heuristic does not consider the amount of requests for a given content, there are many requests for contents whose replicas have been removed on previous time periods. Nevertheless, this behaviour is much more evident on the non-optimized instances.



(a) 10 servers

(b) 20 servers

(c) 30 servers

(d) 50 servers

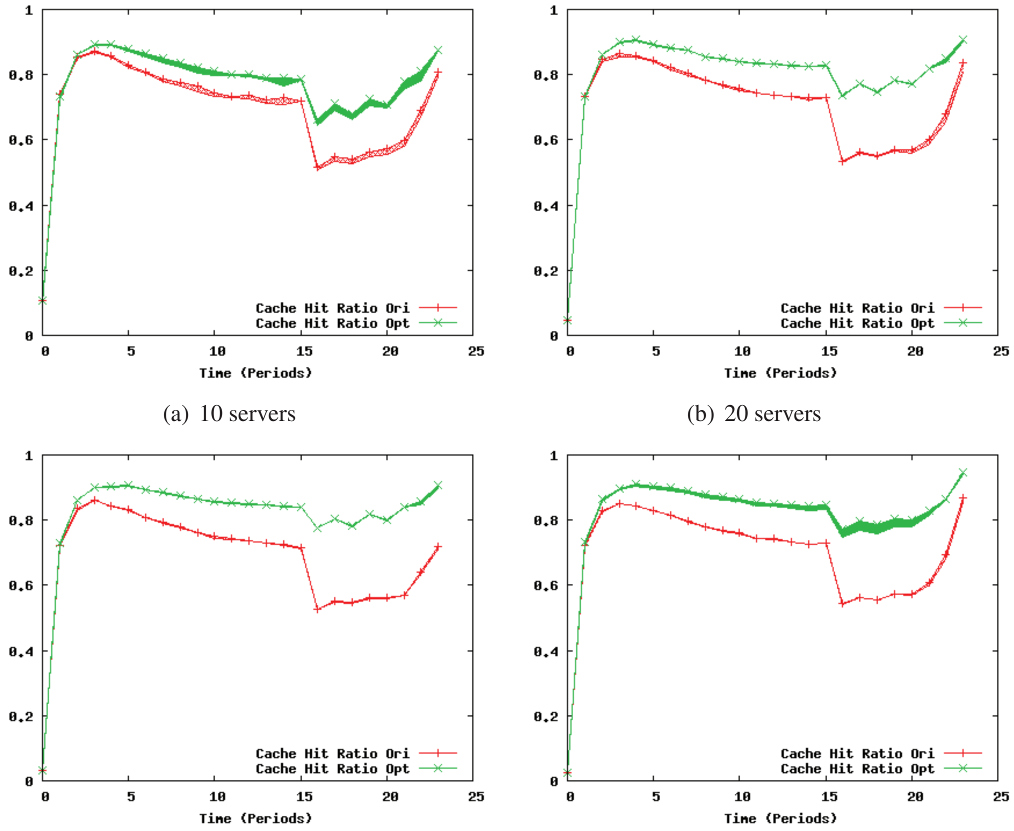
Figure 4. Delivery Costs over time comparison

7 Conclusion and Future Works

On this paper, we have formulated a model to solve the SCAP statically for CDNs with a full-mesh topology. We have presented results that show how a CDN can benefit from a optimized storage capacity allocation among it servers. Reducing its replication and delivery costs is of major importance for CDN providers, as this will reduce their operational costs and improve the satisfaction level of their clients. Also, an improved cache hit ratio on the local server avoids unnecessary transit of data on the networks and ensures that the CDN client will receive the requested content from the closer server.

A granular allocation unit model makes sense on a future Internet scenario where CDN servers are implemented as virtual machines. In such scenario, allocating the storage capacity with high granularity is not only possible, but desirable to avoid the waste of hardware resources. When working on virtual machines scenarios, it is also possible to implement a dynamic approach to storage allocation, where the storage capacity of each CDN server can be increased or decreased on demand. This approach would certainly lead to improved results, as it would be easier to accommodate a sudden increase on the demand for contents without increasing the delivery cost. It is also possible to easily use the high granularity storage capacity allocation results to select the most suited available servers.

As a future work, we plan to improve the SCAP mathematical model in order to contemplate economic costs and benefits related to the CDN business. On this work we have demonstrated that an statically optimized storage capacity allocation can reduce the



(a) 10 servers

(b) 20 servers

(c) 30 servers

(d) 50 servers

Figure 5. Cache Hit Ratio over time comparison

amount of data replicated on a CDN and also can reduce the client perceived network delay while retrieving contents. We believe that by analysing potential economic gains and costs of a CDN provider, we can build a mathematical model that could determine if it is worth for a CDN provider to build an infrastructure to deliver content to a remote site, while determining how much storage capacity must be allocated on each potential point of presence of the CDN.

Acknowledgments

The authors would like to thank Faperj, CNPq and Petrobras for their support on the development of this work.

References

Adamic, L. A. and Huberman, B. A. (2002). Zipf's law and the Internet. *Glottometrics*, 3:143–150.

Khan, S. U., Maciejewski, A. A., and Siegel, H. J. (2009). Robust cdn replica placement techniques. In *Proceedings of the 2009 IEEE International Symposium on Parallel&Distributed Processing*, pages 1–8, Washington, DC, USA. IEEE Computer Society.

Krishnan, P., Raz, K. D., and Shavitt, Y. (2000). The cache location problem. *IEEE/ACM Transactions on Networking*, 8:568–582.

Laoutaris, N., Zissimopoulos, V., and Stavrakakis, I. (2005). On the optimization of storage capacity allocation for content distribution. *Comput. Netw.*, 47(3):409–428.

Li, B., Deng, X., Golin, M. J., and Sohraby, K. (1998). On the optimal placement of web proxies in the internet: The linear topology. In *HPN '98: Proceedings of the IFIP TC-6 Eigth International Conference on High Performance Networking*, pages 485–495, Deventer, The Netherlands, The Netherlands. Kluwer, B.V.

Li, B., Golin, M., Italiano, G., Deng, X., and Sohraby, K. (1999). On the optimal placement of web proxies in the internet. In *INFOCOM '99. Eighteenth Annual Joint Conference of the IEEE Computer and Communications Societies. Proceedings. IEEE*, volume 3, pages 1282 –1290 vol.3.

Li, W., Chan, E., Wang, Y., Chen, D., and Lu, S. (2008). Cache placement optimization in hierarchical networks: analysis and performance evaluation. In *NETWORKING'08: Proceedings of the 7th international IFIP-TC6 networking conference on AdHoc and sensor networks, wireless networks, next generation internet*, pages 385–396, Berlin, Heidelberg. Springer-Verlag.

Neves, T. A., Drummond, L. M. A., Ochi, L. S., Albuquerque, C., and Uchoa, E. (2010). Solving replica placement and request distribution in content distribution networks. *Electronic Notes in Discrete Mathematics*, 36:89–96.

Qiu, L., Padmanabhan, V., and Voelker, G. (2001). On the placement of web server replicas. In *INFOCOM 2001. Twentieth Annual Joint Conference of the IEEE Computer and Communications Societies. Proceedings. IEEE.*

Rajkumar Buyya, Mukaddim Pathan, A. V. (2008). *Content Delivery Networks*. Springer.

Shaikh, A., Tewari, R., and Agrawal, M. (2001). On the effectiveness of dns-based server selection. In *In Proceedings of IEEE Infocom*.

Wang, L., Pai, V., and Peterson, L. (2002). The effectiveness of request redirection on cdn robustness. *SIGOPS Oper. Syst. Rev.*, 36:345–360.

Wu, J. and Ravindran, K. (2009). optimization algorithms for proxy server placement in content distribution networks. In *IM '09: Integrated Network Management-Workshops*, pages 193–198, New York, NY, USA.