# An architecture for selective caching and content distribution as an on-demand service for communities

Università di Napoli Federico II
Via Claudio 21, 80125 Napoli, Italy

**Vittorio Manetti**

COMICS Lab, Dipartimento di Informatica e Sistemistica
email: vittorio.manetti@unina.it

## Introduction

A community can be considered like a collection of users with the same interests; if we are talking about Internet communities, users of the same community probably need the same Internet contents. By definition, community users can be located in the same zone and they can share the same network infrastructure in order to access Internet contents; users are often collected in clusters. As an instance, we can consider students from the same University like a community of users. The contents given to users are supplied by third parties, generic Content Providers, and they are not managed or controlled by community. The Internet contents are often required with very stringent Quality of Services (QoS) features, and usually the Content Providers have the task to solve this kind of problems. In the scenario we are depicting, the community by itself assures high performance in terms of QoS regarding the content distribution. This kind of service is linked with the implementation of *caching mechanisms* that can decrease the latency perceived by a user when he is accessing to the requested content. We propose this solution in order to allow the community to select contents for which this service may be activated *on-demand*. Caches are provided by ISPs or other third parties as a service, or alternatively can be part of the community network infrastructure. In the first case, we have to take into account the cache management fee imposed by the Service Providers.
The main goal of our work is, on one hand, to define an objective function that has to determine the optimal placement of content replicas in a set of candidate web caches, considering a set of cost parameters. On the other hand, we have to realize a network infrastructure for content distribution to user communities. Considering both tasks, we can imagine by now two distinct scenarios: the first one for web contents delivery, the second one for multimedia flow streaming.

## Definition of a model for optimal placement of content replicas

In order to determine a model concerning the problem to solve, we proceed by incremental steps. Starting from the *Simple Plant Location* classical model and improving it, we can realize a model for *optimal placement of content replicas*, considering the available budget like a constraint. We would like to exploit this kind of approach: exact solutions for medium-little networks, and heuristics development for big networks. The model is formulated in a way in which a request from a very asset client is too powerful in comparison with the request from a less asset client. Moreover, we have to consider the limited capacity of caches like a constraint.

*S*, set of servers
*J*, set of caches
*K*, set of clients
*d(k)*, requests from client *k* in relation to time unit
*p(j)*, capacity of cache *j* (kB)
*p(cont)*, dimension of content *cont* (kB)
*qi*, request satisfiable by cache *k*
*c(s,j)*, transmission time on link server-cache
*r(ik)*, transmission time on link cache-client
*h(j)*, location fee for cache *j*
*B*, available budget
*x(sj)*, 1 if the link between server *s* and cache *j* is available (0 otherwise)
*z(jk)*, 1 if the cache between cache *j* and client *k* is available (0 otherwise)
*y(j)*, 1 if the cache is localized on node *j*

$$Minc = \sum_{s,j,cont} c_{sj,cont} x_{sj,cont} + \sum_{j,k,cont} d_{k,cont} r_{jk} z_{jk,cont} \qquad (1)$$

s.a.
$\sum_{j,cont} h_j f_j \leq B$
$\sum_k d_{k,cont} z_{jk,cont} \leq q_{j,cont} y_{j,cont}, \forall j \in J, cont \in O$
$\sum_{cont} q_{j,cont} y_{j,cont} \leq t_j, \forall j \in J$
$\sum_{cont} l_{cont} y_{j,cont} \leq p_j, \forall j \in J$
$\sum_s x_{sj,cont} = y_j, \forall j \in J, cont \in O$
$\sum_{j,cont} z_{jk,cont} = 1, \forall k \in K$
$\mathrm{f}_j \geq y_{j,cont}, \forall j \in J, cont \in 0$
$\mathrm{x}_{sj,cont} = 0/1, \forall s \in S, j \in J, cont \in O$
$\mathrm{z}_{jk,cont} = 0/1, \forall j \in J, k \in K, cont \in O$
$\mathrm{y}_{j,cont} = 0/1, \forall j \in J, cont \in O$
$\mathrm{f}_j = 0/1, \forall j \in J$

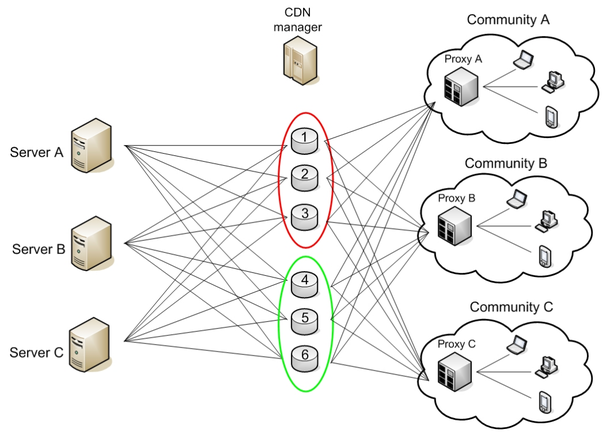
Considering the defined features, we can assure that:

- *the generic request is attended only by open servers*
- *the caches have to satisfy the server requests*
- *client requests have to be served by cache*

We used Xpress to find the best solution for the formulated problem.

## Design and implementation of an architecture for content distribution

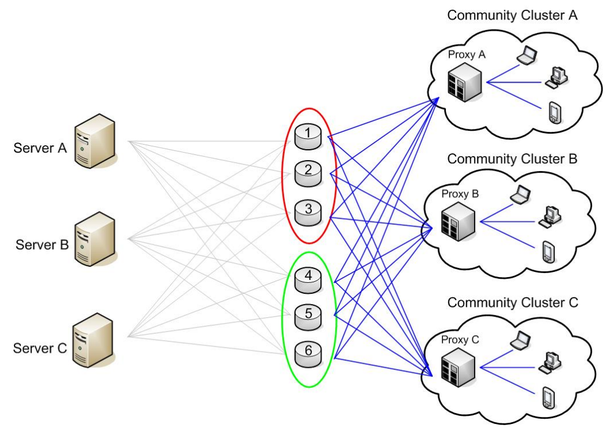
The architecture we have in mind presents new entities with the task to manage content replicas into the cache system in a dynamic and transparent way, in order to optimize the Quality of Experience measured by the generic community user. We introduce a *Proxy* node and a *CDN Manager* node.




## The Proxy node

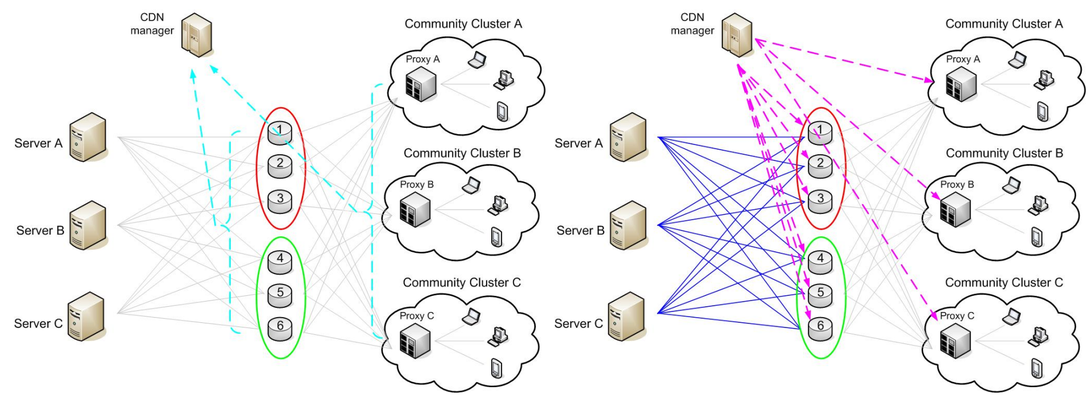
The adopted solution consists in introduce a Proxy node for each cluster of users (as aforementioned, all the users in a cluster share the same network infrastructure). The Proxy has to forward the user request; obviously, this task is based on the content replicas location. The Proxies have to be instructed by CDN Manager on the routing policies to adopt, in order to forward the request on the right cache. The contents are organized in a way in which different caches can provide the same content. There are direct connections between clients and Proxies, and no direct connections between clients and caches; a community Proxy refers to a single cache for a specific object, and it can refer to several caches in the same time in order to satisfy multiple requests of different objects. Each proxy is equipped by a routing table in order to associate a content with the cache that contain it; each entry of this table contains the following fields: *Host*, *Port*, *URL*, *CacheID*.
In the typical scenario:

1. *the Proxy receives a request from a client;*
2. *it queries the routing table to find a connection between the URL with a cache;*
3. *if there is an entry in the routing table associated with the requested URL, the Proxy redirects the request to the specified cache, otherwise it redirects the request to a Content Provider.*




## The CDN Manager node

The CDN Manager has to determine the optimal placement of content into the cache system; this task is based on the objective function previously designed. As we explained above, this function solves the location problem considering a set of cost parameters and metrics opportunely established. The CDN Manager during the run phase has to obtain information needed to compute the objective function, and, based on the achieved results, it has to determine the optimal placement of content replicas into the cache system. The CDN Manager collects information from caches and Proxies, and it deliveries to them information needed to distribute the content replicas. In other words, it computes the optimal way to deploy content into the caches, and the best configuration of the Proxy tables.




## Conclusion and future work

The proposed work is about the design and implementation of a system for optimal placement of multimedia contents for community of users; the model is based on the use of a distributed cache system and on the computation of a predetermined objective function. The work consists, on one hand, in the definition of the objective function, and, on the other hand, in the realization of a network infrastructure. Regarding this second task, we introduce in the architecture two specific nodes and their functionalities: Proxy and CDN Manager.
The proposed model and architecture are still under analysis and in expansion, so it has to be considered like a work in progress. We are considering, for example, the possibility to realize a cache infrastructure organized in a hierarchical manner; moreover, we have in mind to realize a lightweight version of the Proxy in order to allow the installation of this application directly on the clients. Finally, we would like to exploit peer-to-peer technologies to allow the Proxies to cooperate and to distribute the functionalities implemented by CDN Manager between them. In order to validate the effectiveness and the performance of our model, we also have to realize simulations and emulations, exploiting, for instance, a distributed system like PlanetLab.

## References

[1] N. Laoutaris, G. Smaragdakis, K. Oikonomou, I. Stavrakakis, A. Bestavros, *"Distributed Placement of Service Facilities in Large-Scale Networks,"* to appear in IEEE INFOCOM 2007.

[2] N. Laoutaris, V. Zissimopoulos, I. Stavrakakis, *"Joint Object Placement and Node Dimensioning for Internet Content Distribution,"* Information Processing Letters, Vol. 89, No. 6, pp. 273-279, March 2004.

[3] N. Laoutaris, V. Zissimopoulos, I. Stavrakakis, *"On the Optimization of Storage Capacity Allocation for Content Distribution"*, Computer Networks, Vol. 47, No. 3, pp. 409-428, February 2005.

[4] W. Shi, Y. Mao, *"Performance evaluation of peer-to-peer web caching systems"*, Journal of Systems and Software, Volume 79, Pages: 714 - 726, Year of Publication: 2006, ISSN:0164-1212

[5] G. Tsuchida, T. Okino, T. Mizuno, S. Ishihara, *"Evaluation of a replication method for data associated with location in mobile ad hoc networks,"*, ICMU'05, pp. 116–121, 2005.

[6] A. Wierzbicki, *"Models for internet cache location"*, in the 7th Int. Workshop on Web Content Caching and Distribution (WCW), 2002.

[7] A. Vakali, G. Pallis, *"Content delivery networks: Status and trends"*, IEEE Internet Computing, 7(6):68.74, December 2003.

CONTENT Excellence in Content Distribution Network Research