Theoretical Computer Science 540–541 (2014) 82–88

Contents lists available at SciVerse ScienceDirect

Theoretical Computer Science

journal homepage: www.elsevier.com/locate/tcs

Optimal data placement on networks with a constant number of clients

Eric Angel [a], Evripidis Bampis [b], Gerasimos G. Pollatos [c,*], Vassilis Zissimopoulos [c]

[a] IBISC, Université d'Evry Val d'Essone, Evry, France

[b] LIP6, CNRS UMR 7606, Université Pierre et Marie Curie, Paris, France

[c] Department of Informatics and Telecommunications, University of Athens, Greece

ARTICLE INFO

Keywords:
Data placement
Page-placement
Capacity constraints
Dynamic programming

ABSTRACT

We introduce optimal algorithms for the problems of data placement (DP) and page placement (PP) in networks with a constant number of clients each of which has limited storage availability and issues requests for data objects. The objective for both problems is to efficiently utilize each client's storage (deciding where to place replicas of objects) so that the total incurred access and installation cost over all clients are minimized. In the PP problem an extra constraint on the maximum number of clients served by a single client must be satisfied. Our algorithms solve both problems optimally when all objects have uniform lengths. When object lengths are non-uniform we also find the optimal solution, albeit a small, asymptotically tight violation of each client's storage size by εl_{max} where l_{max} is the maximum length of the objects and ε some arbitrarily small positive constant. We make no assumption on the underlying topology of the network (metric, ultrametric, etc.), thus obtaining the first non-trivial results for non-metric data placement problems.

1. Introduction

Peer-to-peer file sharing networks have become one of the most popular aspects of everyday internet usage. Users from all around the globe interact in an asynchronous manner, benefiting from the availability of the desired content in neighboring or more distant locations. The success of such systems stems from the exploitation of a new resource, different from the traditional bandwidth-related resources, namely the distributed storage. Widespread utilization of this new resource is due to the fact that larger capacities have become cheaper, with significantly smaller data access times. Interacting users utilize this resource by installing local storage, replicating popular content and then making it available to neighboring users, thus dramatically decreasing bandwidth consumption needed to access content from the origin servers at which it is available.

A suitable abstract model describing perfectly the aforementioned situation is the *data placement* problem (DP) [1]. Under this model, a set of clients (equivalently users or machines) with an underlying topology is considered and each client has a local amount of storage (cache) installed. Given the set of available objects and the preference that each client has for each object, the objective is to decide a replication scheme, also referred to as a *placement* of objects to local caches so as to minimize the total access cost among all clients and objects. The generalization of this model, under which each client's cache has an upper bound on the number of clients it can serve, is known as the *page placement* problem (PP) [12].

* Corresponding author. Tel.: +30 6948503965.
E-mail addresses: angel@ibisc.univ-evry.fr (E. Angel), Evripidis.Bampis@lip6.fr (E. Bampis), gpollatos@gmail.com, gpol@di.uoa.gr (G.G. Pollatos), vassilis@di.uoa.gr (V. Zissimopoulos).

0304-3975/$ – see front matter
http://dx.doi.org/10.1016/j.tcs.2013.03.025

Table 1
The main known results on data placement problems (*non-uniform lengths with overflow on clients and cache capacities, **non-uniform lengths with overflow on cache capacities only).

	Known results		In this paper	
	Arbitrary *M*		Fixed *M*	
	Metric	No metric	Metric	No metric
Uniform lengths DP	10-approx [1,2,6]	–	Optimal	
Non-uniform lengths DP	10-approx with overflow $l_{\max}$ [2]	–	Optimal with overflow εl_{max}	
Page placement	13-approx [6]*	–	Optimal with overflow** εl_{max}	
Connected DP	14-approx [2]	–	Optimal	
k-median DP	10-approx [1,2]	–	Optimal	

It should be noted that the term replication is used here instead of caching, because under the discussed model, a client cannot change the contents of its local storage without the re-invocation of a replication algorithm. On the contrary, the term caching refers to the process of storing objects locally, coupled with a replacement policy for displacing some of them in favor of others, when the local storage is fully utilized.

Our contributions. We describe optimal algorithms, combining configurations generation and dynamic programming techniques, for the data placement and page placement problems, when the number of clients is constant. This is a natural variation, interesting from both a theoretical and practical point of view ([3,4,7]). Up to now, the only way to tackle these problems was the 10-approximation algorithm of [2] and the 13-approximation algorithm of [6], both designed for the general case and both based on rounding the solution of an appropriate linear program. When object lengths are uniform (or equivalently unit) our algorithm finds the optimum solution in polynomial time. When object lengths are non-uniform, our algorithm returns an optimum solution which violates the capacities of the clients' caches by a small, asymptotically tight additive factor. Our results, summarized in Table 1, can be modified to handle various extensions of the basic problems such as the connected data placement problem ([2]) where object updates are frequent and consistency of all replicas of each object has to be guaranteed and the *k*-median variant of DP where bounds are imposed on the number of maximum replicas allowed for each object. Furthermore, our results are applicable with uniform and non-uniform object lengths and can be employed independently of the underlying topology of the network, thus giving the first non-trivial results for non-metric DP problems.

Related work. The study for the data placement problem over an arbitrary network where all inter-client distances form a metric was initiated in [1] where the authors proved that the problem in the case of objects of uniform length is MAXSNP-hard. They also devised a polynomial 20.5-approximation algorithm based on the rounding of the optimal solution of a suitable linear program. In the case of objects of non-uniform length, the authors proved that the problem of deciding whether an instance admits a solution is NP-complete and provided a polynomial 20.5-approximation algorithm that produces a solution at which the capacity of each client's cache exceeds its capacity in the optimum solution by at most the length of the largest object. The approximation ratio for unit-sized objects was later improved to 10 in [14,2].

Various previous works have also considered variants of the data placement problem in terms of the underlying topology. In [11] the authors consider the case of distances in the underlying topologies that form an ultrametric, i.e. are non-negative, symmetric and satisfy the *strong* triangle inequality, that is $d(i, j) \leq \max\{d(i, k), d(k, j)\}$ for clients i, j, k.The authors consider a simple hierarchical network consisting of three distances between the clients and devise a polynomial algorithm for the case of unit-sized objects by transforming it to a capacitated transportation problem [5]. For the case of general ultrametrics, an optimal polynomial algorithm is given in [9] based on a reduction to the min-cost flow problem.

The *page-placement problem* is an important generalization of the data placement and was proposed and studied in [12]. In this problem, each client has an extra constraint on the number of other clients it can serve, apart from the constraint on the capacity of its cache. In [12], the authors give a 5-approximation algorithm for the problem which violates both client and cache capacity constraints by a logarithmic factor at most. In [6], the logarithmic violation of both capacity constraints was improved to constant with a 13-approximation algorithm. Finally, in [10] and [13] a game-theoretic aspect of the data placement problem is studied, where clients are considered to be selfish agents. In both works, algorithms are provided which stabilize clients in equilibrium placements.

All previous results capture situations where write requests are rarely or never issued for the objects. In [2] the authors consider the case when write requests are common and formulate the *connected data placement* problem, in which it is required that all replicas of an object o are connected via a Steiner tree T_o to a root r_o, which can later be used as a multicast tree. The objective is the minimization of the total incurred access cost and the cost of building the Steiner tree. A 14-approximation algorithm for the problem is given in [2]. This problem is a generalization of the *connected facility location problem* for which the best known approximation ratio is 8.55 [15].