Memory Allocation for Hierarchical Content Distribution Systems

Nikolaos Laoutaris, Vassilios Zissimopoulos, Ioannis Stavrakakis
{laoutaris,vassilis,istavrak}@di.uoa.gr

Department of Informatics and Telecommunications,
University of Athens, 15784 Athens, Greece

I. Introduction

Recent trends toward the improvement of end-user service tend to Supplement the traditional bandwidth-centric internet with a second resource storage capacity (or memory). The addition of storage capacity in network nodes for the caching or replication of popular data objects results in reduced end-user delays, reduced network traffic and improved scalability. The problem of allocating an available storage budget to the nodes of a hierarchical content distribution system is formulated and fast, efficient algorithms are developed for its solution. These algorithms minimize the operating cost of the system and at the same time cater to load balancing and load constraints on the amount of demand that may be serviced by a given node. Strictly hierarchical, as well as hierarchical with peering, request routing models are considered. The work presents an alternative/more generic paradigm for the allocation of memory in distribution networks as opposed to previous formulations that have focused on the selection of location points for the installment of web-site replicas or proxy caches. Compared to web-site replica placement our work can handle content from different web-sites which are also fractionally replicated (most popular objects). Compared to proxy placement we avoid making some assumptions (e.g., known cache hit ratio, same set of objects, thus no co-operation, between different proxies).

II. Contributions

Our work makes the following distinct contributions:

Introduces the idea of provisioning memory using a very small granule as an alternative/extension of known paradigms (mirror placement, proxy placement) and formulates an optimal solution to this problem. The derived solutions provide for a joint optimization of storage capacity allocation and object placement and can be exploited in systems that perform replication (e.g., a CDN) as well as in those that perform caching.

Develops fast efficient heuristic algorithms that approximate closely the optimal performance but can execute very fast, as required by self organizing systems (and as opposed to planning problems that can employ slow algorithms). Moreover these algorithms may be executed incrementally thus obliterating the need for re-optimization from scratch.

Supplements the algorithms for the optimization of storage capacity allocation with the ability to take into consideration load balancing and load constraints on the maximum demand that may be serviced by a node. Load balancing is an important issue in its own right and is jointly addressed here. The presented techniques may be used to avoid overloading some nodes while at the same time underutilizing others, as done by the well known "filtering" effect in hierarchical caches (C. Williamson [1]). The filtering effect is addressed by allocating the storage, as well as the most popular objects, more evenly to the nodes of the hierarchy. Models and studies the effect of request peering, i.e., of the ability to forward request to peer nodes at the same level of the hierarchy as opposed to pure hierarchical systems that only forward requests to upstream ancestor nodes.

III. Conclusions

This work has considered the storage capacity allocation problem and has developed a linear time efficient heuristic algorithm, iGreedy, for its solution. iGreedy and its variants (for load balancing and request peering) have been shown to give good approximations of the optimal solutions, by means of numerical comparison against the bound of the optimal ones (obtained using LP-relaxations).

iGreedy may be used for the derivation of a joint storage capacity allocation and object placement plan to be employed in a system that performs replication, e.g., a CDN. Alternatively, just the derived per node storage capacity allocation may be used for the dimensioning of a hierarchical system that operates under a dynamic caching/replacement algorithm. The provisioning of storage under iGreedy has been compared to that under two empirical methods and a gain has been demonstrated. Additionally, iGreedy has been used for the derivation of numerical results that provide insight into the effects of user request patterns (skewness of demand, homogeneity of demand) on the vertical dimensioning of a hierarchical system.

iGreedy has been modified to cater to load balancing and request peering. Load balancing has been shown to be able to provide for an even spread of load in the hierarchy, sacrificing a small increase in average distance between users and objects, which, however, may be compensated by a substantial reduction of delay as a result of accessing uncongested end-systems. Request peering has been shown to be able to provide for a significantly reduced cost. Request peering may even lead to the halving of the overall cost, when utilizing inexpensive direct peer links.

REFERENCES

[1] C. Williamson, "On filter effects in web caching hierarchies," *ACM Transactions on Internet Technology*, vol. 2, no. 1, pp. 27–77, Feb. 2002.