Cache Optimization Models and Algorithms
Other titles in Foundations and Trends® in Communications and Information Theory

*Group Testing: An Information Theory Perspective*
Matthew Aldridge, Oliver Johnson and Jonathan Scarlett
ISBN: 978-1-68083-596-0

*Sparse Regression Codes*
Ramji Venkataramanan, Sekhar Tatikonda and Andrew Barron

*Fundamentals of Index Coding*
Fatemeh Arbabjolfaei and Young-Han Kim

*Community Detection and Stochastic Block Models*
Emmanuel Abbe
Cache Optimization Models and Algorithms

Georgios S. Paschos
Amazon.com
gpasxos@gmail.com

George Iosifidis
Trinity College Dublin
george.iosifidis@tcd.ie

Giuseppe Caire
TU Berlin
caire@tu-berlin.de

Full text available at: http://dx.doi.org/10.1561/0100000104
Editorial Scope

Topics

Foundations and Trends® in Communications and Information Theory publishes survey and tutorial articles in the following topics:

- Coded modulation
- Coding theory and practice
- Communication complexity
- Communication system design
- Cryptology and data security
- Data compression
- Data networks
- Demodulation and Equalization
- Denoising
- Detection and estimation
- Information theory and statistics
- Information theory and computer science
- Joint source/channel coding
- Modulation and signal design
- Multiuser detection
- Multiuser information theory
- Optical communication channels
- Pattern recognition and learning
- Quantization
- Quantum information processing
- Rate-distortion theory
- Shannon theory
- Signal processing for communications
- Source coding
- Storage and recording codes
- Speech and Image Compression
- Wireless Communications

Information for Librarians

Foundations and Trends® in Communications and Information Theory, 2020, Volume 16, 4 issues. ISSN paper version 1567-2190. ISSN online version 1567-2328. Also available as a combined paper and online subscription.
## Contents

1 **Introduction** 2
   1.1 Historical Background and Scope 2
   1.2 The Content Delivery Network 4
   1.3 Wireless Caching and Beyond 7
   1.4 Structure 8

2 **Content Popularity** 11
   2.1 Introduction to Caching-Related Terms 11
   2.2 Power Law Popularity 13
   2.3 Request Sequences 25
   2.4 Discussion 36

3 **Cache Eviction Policies** 37
   3.1 Performance Under Arbitrary Requests 37
   3.2 Performance Under Stationary Requests 49
   3.3 Online Popularity Learning 59
   3.4 Discussion of Related Work 75

4 **Caching Networks** 78
   4.1 Model and Optimization Variables 78
   4.2 Deployment of Caching Networks 83
   4.3 Bipartite Caching Networks 90
5 Online Bipartite Caching  
  5.1 Background and Model  
  5.2 Problem Statement  
  5.3 Bipartite Supergradient Caching Algorithm  
  5.4 Extensions and Numerical Evaluation  
  5.5 Discussion of Related Work  

6 Asymptotic Laws for Caching Networks  
  6.1 Analysis of Large Caching Networks  
  6.2 Discussion of Related Work  

Acknowledgments  

References
ABSTRACT

Caching refers to the act of replicating information at a faster (or closer) medium with the purpose of improving performance. This deceptively simple idea has given rise to some of the hardest optimization problems in the fields of computer systems, networking, and the Internet, many of which remain unsolved several years after their conception. While a wealth of research contributions exists from the topics of memory systems, data centers, Internet traffic, CDNs, and recently wireless networks, the literature is dispersed and overlapping at times. In this monograph, we take a unifying modeling view: by focusing on the fundamental underlying mathematical models, we re-organize the available material into a powerful framework for performing optimization of caching systems. This way, we aspire to present a solid background for the anticipated explosion in caching research, but also provide a didactic view into how engineers have managed to infuse mathematical models into the study of caching over the last 40 years.
1

Introduction

Storage resources and caching techniques permeate almost every area of communication networks today. In the near future, caching is set to play an important role in storage-assisted Internet architectures, information-centric networks and wireless systems, reducing operating and capital expenditures and improving the offered services. In light of the remarkable data traffic growth and the increasing number of rich-media applications, the impact of caching is expected to become even more profound than it is today. Therefore, it is crucial to design these systems in an optimal fashion, ensuring the maximum possible performance and economic benefits from their deployment. To that end, this monograph presents a collection of detailed models and algorithms, which are synthesized to build a powerful analytical framework for caching optimization.

1.1 Historical Background and Scope

The term cache was introduced in computer systems in 1970s to describe a memory with very fast access but typically small capacity. In computer applications, memory access often exhibits locality, i.e., most requests are related to memory blocks in a specific area known as hot spot.
1.1. Historical Background and Scope

By replicating these spots on a cache, it is possible to accelerate the performance of the entire memory system. One of the most important first problems in this context was to select which memory blocks to replicate in order to maximize the expected benefits; and several key results, e.g., the oracle MIN policy \cite{26}, were developed in that early era of computer systems. Nevertheless, the design of such caching policies remains one of the main challenges in caching systems.

The above caching idea was later applied to the Internet. As the population of users was growing fast in 1990s, the client-server connection model became impractical since all content requests (for web pages, in particular) were routed to few central servers. This was creating server and network congestion, and motivated the idea of using Internet caches. The latter are deployed closer to end users and host carefully selected web pages. Given the content popularity skewness, i.e., the fact that few web pages attract the majority of requests, even small caches can bring impressive performance benefits. Indeed, it soon became clear that web caching can significantly alleviate network congestion and improve the content access time for users. In these interconnected caches the problem of designing optimal caching policies is more intricate, as it requires to decide which files to cache, how to route the content and how to dimension each cache.

The last few years we witness a resurgence of interest in caching in the domain of wireless networks. The expansive growth of mobile video traffic in conjunction with exciting developments — like the use of coding techniques — have placed caching at the forefront of research in wireless communications. There is solid theoretical and practical evidence that memory can be a game-changer in our efforts to increase the effective throughput, and there are suggestions for deploying caches at the network core, the base stations or even at the mobile devices. Similarly, novel services that involve in-network computations, require pre-stored information (e.g., machine learning services), or are bounded by low latency constraints, can greatly benefit from caching. In fact, many caching enthusiasts argue that such services can only be deployed if they are supported by intelligent caching techniques.

Amidst these developments, it is more important than ever to model, analyze and optimize the performance of caching systems. Quite
Introduction

Surprisingly, many existing caching solutions, albeit practical, have not been designed using rigorous mathematical tools. Hence, the question of whether they perform optimally remains open. At the same time, the caching literature spans more than 40 years, different systems and even different research communities, and there is lack of a much-needed unified view on caching models and algorithms. This monograph aspires to fill this gap by presenting the theoretical foundations of caching and the latest conceptual and mathematical advances in this area. It provides detailed technical arguments and proofs, aiming to create a stable link between the past and future of caching analysis, and offer a useful starting point for new researchers. In the remainder of this section, we set the ground by discussing key caching systems and ideas, and explain the organization of this monograph.

1.2 The Content Delivery Network

A milestone in the evolution of caching systems was the deployment of Content Delivery Networks. CDNs typically consist of: (i) the origin server; (ii) the dispersed caches; (iii) the backbone network; and (iv) the points of ingress user traffic. The origin server is often deployed at a remote location with enormous storage capabilities (e.g., a datacenter) and stores all content files a user of this service might request, i.e., the entire content catalog. The caches are smaller servers deployed near the demand points, and are connected with the backbone network. Before CDNs, the users would establish TCP connections with the origin server in order to retrieve the content. In CDNs however, these connections are redirected to caches which serve the requests for these locally replicated files. Modern CDN systems have further evolved to optimize network traffic, offer different levels of Quality of Service (QoS), and increase the robustness of these services by, for example, protecting them from Denial of Service (DoS) attacks. In this monograph we focus on the aspect of content caching, and specifically on the intelligence involved in orchestrating the caching operations.

An iconic CDN system is the “Akamai Intelligent Platform”, see [181] for a detailed description. Akamai was one of the most prominent CDN providers in the booming Internet era of 2000s, and by 2019 was
1.2. The Content Delivery Network

responsible for delivering 20% approximately of the Internet traffic. Its 216 K caching servers are dispersed at network edges offering low-latency content access around the globe. The Akamai model was designed to intercept HTTP traffic using a DNS redirect: when a user wants to open a website with the HTTP protocol, it would first contact the local DNS server to retrieve the IP of the origin server. The intelligent platform replaces the DNS entry with the IP of an Akamai cache containing the requested content, and hence the HTTP request is eventually served by that cache. The intelligent operations are handled by the mapping system, which decides where to cache each content file, and accordingly maps DNS entries to caches. Although the mapping system is effectively deciding the placement of content, the local caches are also operated with reactive policies such as the famous LRU and its variants.

Benefits of Caching

The replication of few popular contents can significantly reduce the traffic at the backbone network. When a content is available at a nearby cache, the respective requests are redirected to that cache instead of the origin server (an event called hit). Therefore, caches are often scattered around the network to minimize the geodesic distance, or network hops, from the users. Previous research has investigated solutions for the optimal placement of servers, e.g., [201], and the sizing of cache storage, called cache dimensioning [135]. Since more hits mean less network traffic, an important criterion for deploying caches is the increase of the cache hit ratio. Pertinent optimization problems in this context include the choice of eviction policies, i.e., the dynamic selection of the contents that are evicted from an overflowing cache, and the strategic content placement for enabling cache collaboration [24].

Another benefit of caching is latency reduction, i.e., the decrease of elapsed time between the initiation of a request and content delivery. Typically, the latency improvement is attributed to cutting down propagation latency. Packets traversing a transcontinental link, for example, experience latency up to 250 msec due to speed-of-light limitations [218]. Given that each TCP connection involves the exchange of several messages, it might actually take seconds before a requested content is
delivered over such links. These large latencies are very harmful for e-commerce and other real-time applications, and their improvement has been one of the main market-entry advantages of CDNs. Indeed, when users retrieve contents from a nearby cache, the geodesic distance is greatly reduced and so is the propagation time that hinders the content delivery. Nevertheless, latency optimization in caching systems is an intricate task and there are some notable misconceptions.

Firstly, in most applications latency effects smaller than 30 msec do not impact the user experience. Hence, one needs to be cautious in increasing the infrastructure costs in order to deliver content faster than this threshold. In other words, when it comes to latency, a single local cache often suffices to serve a large metropolitan area. Secondly, regarding video content delivery, the latency requirements apply only to the first video chunks and not to the entire file. Delivering fast the first chunks and then exploiting the buffer capacity at the user side is often adequate for ensuring smooth reproduction, even if the later video segments are delivered with higher latency. Finally, several low latency applications, such as reactive virtual reality, vehicular control, or industrial automation, cannot typically benefit from caching since their traffic is not reusable. However, we stress that there are scenarios where one can exploit caching (e.g., using proactive caching policies) in order to boost the performance of such demanding services.

Another important effect of web caching is that it balances the server load. For example, the Facebook Photo CDN leverages web browser caches on user devices, edge regional servers, and other caches in order to reduce the traffic reaching the origin servers. Notably, browser caches serve almost 60% of traffic requests, due to the fact that users view the same content multiple times. Edge caches serve 20% of the traffic (i.e., approximately 50% of traffic not served by browser caches), and hence offer important off-network bandwidth savings by serving locally the user sessions. The remaining 20% of content requests are served at the origin, using a combination of slow back-end storage and a fast origin-cache [116]. This CDN functionality shields the main servers from high load and increases the scalability of the architecture. Note that the server load is minimized when the cache hits are maximized, and hence the problem of server load minimization is equivalent to
cache hit maximization. Therefore, in the remaining of this monograph we will focus on hit maximization, as well as bandwidth and latency minimization.

### 1.3 Wireless Caching and Beyond

Caching has also been considered for improving content delivery in wireless networks, see [191] and references therein. There is consensus that network capacity enhancements by means of improving physical layer access rates or through denser deployment of base stations is a costly approach and outpaced by the fast-increasing data traffic [60]. Caching techniques promise to fill this gap, and several interesting ideas have been suggested to this end: (i) deep caching at the evolved packet core (EPC) in order to reduce content delivery delay [242]; (ii) caching at the base stations to alleviate congestion in their throughput-limited backhaul links [103]; (iii) caching at the mobile devices to leverage device-to-device communications [101]; and (iv) coded caching for accelerating transmissions over a broadcast medium [163].

Furthermore, techniques that combine caching with coding demonstrate revolutionary goodput scaling in bandwidth-limited cache-aided networks. This has motivated researchers to revisit the fundamental question of how memory “interacts” with other resources. The topic of coded caching started as a powerful tool for broadcast mediums, and led towards establishing an information theory for memory. Similarly, an interesting connection between memory and processing has been identified [154], creating fresh opportunities for improving the performance of distributed and parallel computing systems. These lines of research have re-stirred the interest in joint consideration of bandwidth, processing and memory resources.

At the same time, the advent of technologies such as Software-Defined Networking (SDN) and Network Function Virtualization (NFV) create new opportunities for leveraging caching in wired and wireless networks. Namely, they enable the fine-grained and unified control of storage, computing and network bandwidth, and support the flexible deployment of in-network caching services. This gives rise to the new concept of content-centric network architectures that aim to use storage
and caching as a means to rethink the Internet operation. Similarly, new business models are emerging as new players are entering the content delivery market. Service providers like Facebook are acquiring their own CDNs, network operators deploy in-network cache servers to reduce their bandwidth expenditures, and content providers like Google, Netflix, and Amazon use caches to replicate their content world-wide. Interestingly, smaller content providers can buy caching resources on the cloud market to instantiate their service just in time and space. These novel concepts create, unavoidably, new research questions for caching architectures and the caching economic ecosystem, and one of our goals is to provide the fundamental underlying mathematical theories that can support research in these exciting directions.

1.4 Structure

In this subsection we provide a quick summary of the monograph, serving both as a warm-up for reading it, as well as a map with directions to specific information.

We begin in Section 2 with a detailed treatment of content popularity models, a crucial factor shaping the performance of caching policies. We first explain the power-law popularity model and how we can infer its parameters from a dataset, and then use it for the purpose of cache dimensioning. We then define the Independence Reference Model (IRM) for describing a request generation process. IRM is a widely used model for caching analysis because of its tractability, but it has limited accuracy since it fails to capture two observed correlation effects of request processes, namely temporal and spatial locality. We discuss state-of-the-art mathematical models which are more accurate than IRM in that respect, but also more difficult to analyze. For the case of temporal correlations, we provide the optimal rule for popular/unpopular content classification that maximizes cache performance.

In Section 3 we explore the class of online eviction policies, also known as replacement policies. A cache receives requests and a rule must be employed for evicting a content when the cache overflows. The design of such policies is an equally challenging and important problem, and we present the main pertinent results. We begin with the case
of arbitrary requests, for which an oracle policy, known as “Belady”, achieves the maximum number of hits under any request sequence. This policy requires knowledge of future requests, and therefore is useful only as a benchmark. Using the Belady policy we prove that the “Least Recently Used” (LRU) rule provides the best competitive performance among all online policies, i.e., those that do not have information about future requests. Then, for stationary IRM requests the “Least Frequently Used” (LFU) rule is the optimal, as it estimates the (assumed static) content popularity using the observed frequencies. We also study the characteristic time approximation, with which we obtain the performance of LRU for stationary requests, as well that of Time To Live (TTL) caches which allow to optimally tune different content hit probabilities. Last, we depart from the stationary assumption and take a model-free approach inspired by the Machine Learning framework of Online Convex Optimization (OCO). We present an adaptation of the Zinkevich’s online gradient policy to the caching problem, and show that it achieves the optimal regret, i.e., the smallest possible losses with respect to the best static cache configuration with future knowledge.

In Section 4 we study caching networks (CNs), i.e., systems where multiple caches are interconnected via a network. We focus on proactive caching policies which populate the caches based on estimated demand. In CNs, the designer needs to decide where to cache each content file (caching policy), how to route the contents from caches to the requesters (routing policy), and also the capacity of the caches and network links. Therefore, we start the section by explaining this general CN design and management problem. This is a notoriously hard problem, that cannot be solved optimally for large CNs and content catalogs. To gain a better understanding into the available solution methodologies, we survey a number of important subproblems: (i) the cache dimensioning problem where we decide where to place storage in the network; (ii) the content caching in bipartite and tree graphs; and (iii) the joint content caching and routing problem in general graphs. Although these are all special cases of the general CN problem, they are governed by significantly different mathematical theories. Therefore, our exposition in this section serves to clarify where each mathematical theory applies best, and how to get a good approximate guarantee for each scenario.
In the following Section 5 we take an approach that combines the two previous sections. We study a CN where the content popularity is unknown, and therefore the goal is to design a joint caching and routing policy which at the same time learns the content popularity, decides in an online manner which files to cache, and in a reactive fashion how to route the contents to requesters. These results generalize the OCO-based policy that was introduced in Section 3 for a single cache. We show that a policy that takes a step in the direction of a subgradient of the previous slot’s utility function can provide “no regret” in the CN scenario as well. This means that the designed policy gradually learns to match the performance of the optimal static policy, a theoretical result that is validated through trace-driven numerical experiments. The analysis in this section is general enough to account for changes in the network structure and cache reconfiguration costs.

In the last Section 6 we examine a very large (scaling) network of caches, arranged in a square grid. For this special case, we show that it is possible to relax the original combinatorial problem and obtain a relaxed solution via convex optimization. Then it can be shown that the relaxed solution is of the same order of performance with the actual integral, hence we can use it to understand the scaling performance of a large caching network. The section includes detailed results about the sustainability of networks aiming to deliver content in different regimes of: (i) network size; (ii) catalog size; and (iii) cache size.
References


References


References


References


<table>
<thead>
<tr>
<th>References</th>
</tr>
</thead>
</table>


References


References


Full text available at: http://dx.doi.org/10.1561/0100000104


