Modélisation de performance des caches basée sur l'analyse de données

Luis Felipe Olmos Marchant

NNT : 2016SACLX008

THÈSE DE DOCTORAT

DE L'UNIVERSITÉ PARIS-SACLAY

préparé à

L'École Polytechnique

ÉCOLE DOCTORALE N° 574

École Doctorale de Mathématiques Hadamard (EDMH)

Spécialité de doctorat : Mathématiques aux interfaces

par

Luis Felipe OLMOS MARCHANT

A Data Driven Approach for Cache Performance Modeling

Thèse présenté et soutenue à Châtillon, le 30 mai 2016

Composition du jury :

M. Erwan LE PENNEC	Président	École Polytechnique
M. Thomas BONALD	Rapporteur	Télécom ParisTech
M. Moez DRAIEF	Rapporteur	Imperial College London
Mme Giovanna CAROFIGLIO	Examinatrice	Cisco Systems
M. Emilio LEONARDI	Examinateur	Politecnico di Torino
M. Laurent MASSOULIÉ	Examinateur	INRIA
M. Carl GRAHAM	Directeur de thèse	École Polytechnique
M. Bruno KAUFFMANN	Codirecteur de thèse	Orange

Acknowledgments

In the collective unconscious, a thesis is still considered a "one-man" work. However, any person who has done research today knows very well that it is a collective effort. This dissertation is no exception and I am thus indebted to the many persons that contributed to it.

First and foremost, I would like to acknowledge my advisors Carl Graham and Bruno Kauffmann. Since this thesis lies at the boundary of mathematics and computer science, their complementary knowledge provided an invaluable guide for its development. Thank you for your constant encouragement and for believing in me.

I am deeply indebted to Alain Simonian, the "unsung hero" of this work, as he effectively acted as a third advisor. His knowledge, rigor and patience was key for this work.

I sincerely thank Thomas Bonald and Moez Draief for the time they dedicated to review my manuscript and for the constructive remarks they made to improve it. I thank as well Giovanna Carofiglio, Erwan Le Pennec, Emilio Leonardi and Laurent Massoulié for having accepted being part of my thesis examination committee.

I had enriching technical discussions with Christian Tanguy, Claudio Imbrenda, Eric Gourdin, Erwan Le Pennec, François Roueff, Leonce Mekinda, Nicaise Fofack, Phillipe Olivier and Yannick Carlinet. The time you have dedicated me has indeed influenced this work, thank you very much.

I am truly grateful to Fabrice Guillemin, head of the "Flexible Network Control" project, for making this thesis possible and for his constant aid. Many thanks as well to Nabil Benameur, Prosper Chemouil and Adam Ouourou for their support and the opportunities they have given me in these years.

Special thanks to Nassera Naar and Régine Angoujard for their administrative support and for always

shedding light in the bureaucratic jungle.

Happily for me (and my health) these three years have been not only about research. I am lucky to have had a good share of leisure moments with many persons.

I thank the people who were at "Traffic Resources Modeling" team from 2013 through 2016 for the day-to-day conviviality moments, conversations and jokes. Thank you Bobby, David, Emmanuel, Fadi, Florence, Ghida, Guanglei, Jean-Baptiste, Luca, Mina, Nancy, Paul, Pierre, Raluca, Thomas, Thibaut and Yassine.

I shared many laughs with the unofficial “Club Espagnol” of Wednesdays at Orange. ¡Gracias Ana María, Antonio, Belén, Daniel, José, Gema y Rocío!

During these three years I have shared precious moments (and beers!) with old and new friends. Thank you Ana, Aser, Celia, Christine, Cristóbal, Daniela, David, Eric, Giorgio, Jerónimo, Kamila, Magdalena, Mathieu, Miraine, Nicolás, Pablo, Pedro, Sebastián D., Sebastián G, Sofía, Teja, Teresa and Valeska.

I thank my family for all supporting me from far away in Chile and for always believing in me.

Last but not least, I thank Andrea, for her support, patience and love.

Abstract

The need to distribute massive quantities of multimedia content to multiple users has increased tremendously in the last decade. The current solution to this ever-growing demand are *Content Delivery Networks*, that handle nowadays the majority of multimedia traffic by means of a distributed architecture. This distribution problem has also motivated the study of new solutions such as the *Information Centric Networking* paradigm, whose aim is to add content delivery capabilities to the network layer by decoupling data from its location. In both architectures cache servers play a key role, allowing efficient use of network resources for content delivery. As a consequence, the study of cache performance evaluation techniques has found a new momentum in recent years.

In this dissertation, we propose a framework for the performance modeling of a cache ruled by the *Least Recently Used* (LRU) discipline. Our framework is data-driven in the sense that, in addition to the usual mathematical analysis, we address two additional data-related problems: the first one is to propose a model that is a priori both simple and representative of the essential features of the measured traffic. The second one is the estimation of the model parameters starting from traffic traces. The contributions of this thesis concerns each of the above tasks.

For our first contribution, we propose a parsimonious traffic model featuring a document catalog evolving in time. We achieve this by allowing each document to be available for a limited (random) period of time. To make a sensible proposal, we apply the "semi-experimental" method to real data. These "semi-experiments" consist in two phases: first, we randomize the traffic trace to break specific dependence structures in the request sequence; secondly, we perform a simulation of a LRU cache with the randomized request sequence as input. For a candidate model, we refute an independence hypothesis if the resulting hit probability curve differs significantly from that obtained from original trace. With the insights obtained, we refute the widely used *Independent Reference Model* (IRM) for our data and

propose a traffic model based on Poisson cluster point processes.

Our second contribution is a theoretical estimation of the cache hit probability for a generalization of the latter model. For this objective, we use the Palm distribution of the arrival process to set up a probability space whereby a document can be singled out for the analysis. In this setting, we then obtain an integral formula for the average number of misses. Finally, by means of a scaling of system parameters, we obtain an asymptotic expansion for the latter integral with large cache size. This expansion quantifies the error of a widely used heuristic in the literature known as the "Che approximation", thus providing both a justification and an extension for the considered class of processes.

Our last contribution concerns the estimation of the model parameters. We tackle this problem in the case of the simpler IRM model. By considering its parameter (a popularity distribution) to be a random sample, we implement a Maximum Likelihood method to estimate it. This method allows us to seamlessly handle the censor phenomena occurring in traces. By measuring the cache performance obtained with the resulting model, we show that this method provides a more representative model of data than typical ad-hoc methodologies.

Résumé Étendu

La nécessité de distribuer des quantités massives de contenus multi-média à un nombre croissant d'utilisateurs s'est accrue au cours de la dernière décennie. La solution actuelle pour cette demande en croissance constante est fourni par les systèmes connues sous le nom de *Content Delivery Networks*, qui gèrent actuellement la majorité du trafic multi-média en utilisant une architecture distribuée. Ce problème de distribution a également motivé l'étude de nouvelles solutions tel que celui proposé par l'*Information Centric Networking*, dont l'objectif est d'ajouter des capacités de livraison de contenus à la couche réseau, moyennant un découplage des données et de leur localisation. Dans ces deux architectures, les serveurs cache jouent un rôle clé, en permettant un usage efficace des ressources de réseau pour la distribution de contenus. En conséquence, l'étude des techniques pour l'évaluation des performances des serveurs cache a trouvé un nouvel élan ces dernières années.

Dans cette thèse, nous proposons un cadre complet pour la modélisation des performances d'un cache utilisant la politique de remplacement *Least Recently Used* (LRU). Notre cadre considère, outre l'analyse mathématique, deux procédures qui relient les données au modèle : Dans la première procédure, nous proposons un modèle simple qui est a priori représentatif des caractéristiques essentielles du trafic mesuré; dans la deuxième nous estimons les paramètres du modèle à partir des traces de trafic. Les contributions de cette thèse concernent chacune des procédures mentionnées. Dans la suite, nous décrivons succinctement chacune de nos contributions.

Proposition d'un nouveau modèle de trafic

Dans notre première contribution, nous proposons un modèle de trafic parcimonieux (i.e. à petit nombre de paramètres) qui décrit un catalogue évoluant dans le temps. Pour effectuer un choix judicieux du modèle, nous appliquons la méthode dite « semi-expérimentale » en utilisant deux jeux des données réelles. Ces données sont des traces de trafic provenant des systèmes des caches du réseau de l'opérateur Orange. Les semi-expériences menées consistent en deux étapes :

1. d'abord, nous randomisons la trace en cherchant à briser d'éventuelles structures de dépendance stochastique;
2. ensuite, nous simulons un cache LRU avec la séquence de requêtes randomisé comme ci-dessus.

Par la suite, pour tout modèle candidat, nous réfutons une hypothèse d'indépendance si la courbe de probabilité de «hit» diffère significativement de celle obtenue avec la trace initiale. Nous avons conduit des semi-expériences visant à réfuter les hypothèses suivantes

(i) Indépendance totale des instants d'arrivées des requêtes;

(ii) Indépendance des instants de parution des documents dans le catalogue;

(iii) Indépendance des instants de requête d'un document donné.

De plus, nous cherchons l'échelle de temps ou l'hypothèse (i) commence à être significative.

Les résultats de ces semi-expériences nous conduisent à réfuter la modélisation de nos données au moyen de l'« Independent Reference Model » (IRM). Bien que ce modèle soit très utilisé dans la littérature du domaine, l'échelle de temps de nos données est trop longue pour que les hypothèses d'independance qu'il exige restent valides.

Néanmoins, les résultats de ces expériences nous mènent naturellement à proposer un modèle de trafic basé sur des processus de type « cluster » poissoniens. Ce modèle consiste en deux couches (voir Figure 1):

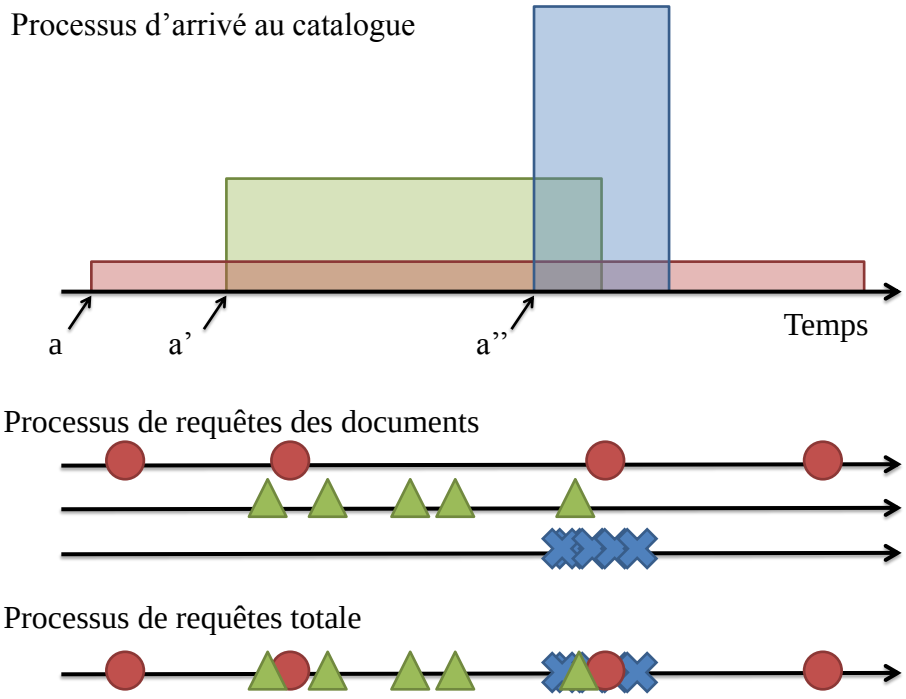


Figure 1: *Une réalisation du processus d'arrivée au catalogue et requêtes.* ***En haut:*** *Les fonctions « boîtes » représentent la durée et la popularité de chaque document par la hauteur et la longueur respectivement.* ***En bas:*** *Une réalisation des processus de requêtes de documents. Leur superposition génère le processus de requêtes total.*

- La première couche modélise les instants d'apparition des documents dans le catalogue. Nous considérons que ces instants sont générés par un processus de Poisson homogène de taux γ.
- Dans la deuxième couche, qui dépend de la première, se trouvent les requêtes vers chaque document. Ces requêtes sont modélisés par un processus de Poisson homogène de durée finie qui commence à un instant d'arrivée défini dans la première couche. L'intensité et la durée de ces processus est aléatoire et fixée pour chaque document.

La superposition des deux couches génère le processus total de requêtes.

Ce modèle a une structure suffisamment riche pour représenter un catalogue de documents évoluant dans le temps et assez simple pour permettre une analyse mathématique de la probabilité de succés ou «hit probability» qui est la mesure de performance que nous étudions dans cette thèse.

Ces travaux on été présentés à la conférence ITC 2014 en collaboration avec Bruno Kauffmann, Alain Simonian et Yannick Carlinet [3].

Estimation de la probabilité de hit

Notre deuxième contribution est une estimation théorique de la probabilité de hit pour une généralisation du modèle de cluster proposé précédemment. Au lieu d'utiliser une fonction «boîte» pour modéliser l'intensité de requêtes d'un document, nous considérons une fonction aléatoire positive λ quelconque, presque sûrement intégrable par rapport au temps.

Dans la première partie de notre analyse, nous utilisons la distribution de Palm du processus global pour construire un espace de probabilité où un document peut être séparé et analysé de façon indépendante du reste des documents. Ce cadre probabiliste nous permet d'étudier le comportement d'un document « moyen ». En particulier, grâce à la nature de la discipline LRU, cette décomposition est très adaptée pour estimer la probabilité de hit sous cette politique.

En travaillant sous la mesure de Palm, nous obtenons une formule intégrale pour le nombre moyen de requêtes «miss» μ_C pour un cache de taille C (notons que cette quantité nous permet d'en déduire immédiatement la probabilité de hit). La formule est donnée par

$$\mathbb{E}[\mu_C] = \mathbb{E}[m(T_C)]$$

où T_C représente le temps de sortie d'un objet dans le cache et $m(t)$ est donné par

$$m(t) = \mathbb{E}\left[\int_0^\infty \lambda(u) e^{-(\Lambda(u+t)-\Lambda(u))}\,\mathrm{d}u\right], \quad t \geq 0\,.$$

Cette dernière quantité n'est autre que le nombre moyen de «miss» dans un cache qui élimine un document après t unités de temps s'il ne reçoit pas une nouvelle requête dans cette durée.

Pour rendre applicable cette formule intégrale, nous appliquons un « scaling » des paramètres du système en supposant la taille C du cache grande et que le taux d'arrivée γ des documents proportionnel

à C. Ceci nous permet d'obtenir un développement asymptotique de $\mathbb{E}[\mu_C]$ sous la forme

$$\mathbb{E}[\mu_C] = m(t_\theta) + \frac{e(t_\theta)}{C} + o\left(\frac{1}{C}\right)$$

ou le terme d'erreur $e(t_\theta)$ ne dépend que de la fonction m et ses dérivés et t_θ est appelé «temps caractéristique» du cache pour la charge $\theta = C/\gamma$. Le temps caractéristique t_θ est donné par l'équation

$$t_\theta = M^{-1}(\theta) \quad ; \quad M(t) = \int_0^t m(s)\mathrm{d}s.$$

Ce terme « temps caractéristique » n'est pas choisi au hasard: le développement quantifie l'erreur de une heuristique très utilisé dans la littérature connue comme l'«Approximation de Che» qui dépend aussi d'un temps caractéristique similaire. Nous démontrons que ce temps caractéristique coïncide avec celui de notre développement et que l'approximation équivaut à tronquer notre développement à l'ordre 0. Par cette résultat, nous justifions et étendons cette heuristique pour la classe des processus considérée.

Une version preliminaire de ces travaux a été présenté dans l'article ITC26 déjà mentionné [3]. La version génerale et complète a été soumis au journal « Stochastic Systems » en collaboration avec Carl Graham et Alain Simonian et il est à ce jour en cours de révision [1] .

Estimation de paramètres

La dernière contribution de nos travaux concerne l'estimation des paramètres du modèle. Nous abordons ce problème dans le cas le plus simple du modèle IRM.

Les paramètres de ce modèle de trafic sont la taille du catalogue K et la distribution de popularité $R_1, R_2, \ldots, R_K$. Nous montrons que l'estimation de ces paramètres avec la méthode du Maximum de Vraisemblance est pratiquement infaisable. Pour nous placer dans un cadre où nous pouvons appliquer cette méthode, nous modifions le modèle IRM en considérant que la distribution de popularité n'est pas constante mais est un échantillon aléatoire tiré d'une distribution inconnue g. Nous appelons ce modèle « IRM-Mixed » (IRM-M) car la distribution du nombre de requêtes par document suit alors une loi de Poisson composée avec avec distribution «mélangeante» g. Nous remarquons que ce modèle peut

être considéré comme intermédiaire entre IRM et le modèle de cluster que nous avons proposé, où le nombre de requêtes de chaque document suit aussi une loi de Poisson composée.

Dans le cas du modèle IRM-M, l'estimation par la méthode du Maximum de Vraisemblance est alors traitable et nous implémentons un algorithme pour l'appliquer. Un avantage additionnel de cette méthode est qu'il permet de traiter de façon transparente les phénomènes de censure qui interviennent dans les données: par exemple, les documents sans requêtes ne laissent pas des trace de trafic, et donc une partie de la distribution du nombre de requêtes n'est pas observable.

Nous évaluons notre méthode suivant trois axes:

(i) l'estimation de la distribution mélangeante g qui représente la popularité de chaque document;

(ii) l'estimation de la distribution mélangée Poisson(g) qui représente le nombre de requêtes par document;

(iii) l'estimation de la probabilité de hit d'un cache LRU

Nous démontrons que notre méthode fourit un modèle moins biaisé et plus représentatif des données. Ces travaux ont été présentés à la conférence VALUETOOLS 2015 en collaboration avec Bruno Kauffmann [2] .

Contents

Chapter 1

Introduction

1.1 What is Caching?

In computer systems and networks, caching is a trade-off between storage and transfer resources that increase the system efficiency. For many use-cases, caching actually makes the difference between a feasible and a non-feasible system.

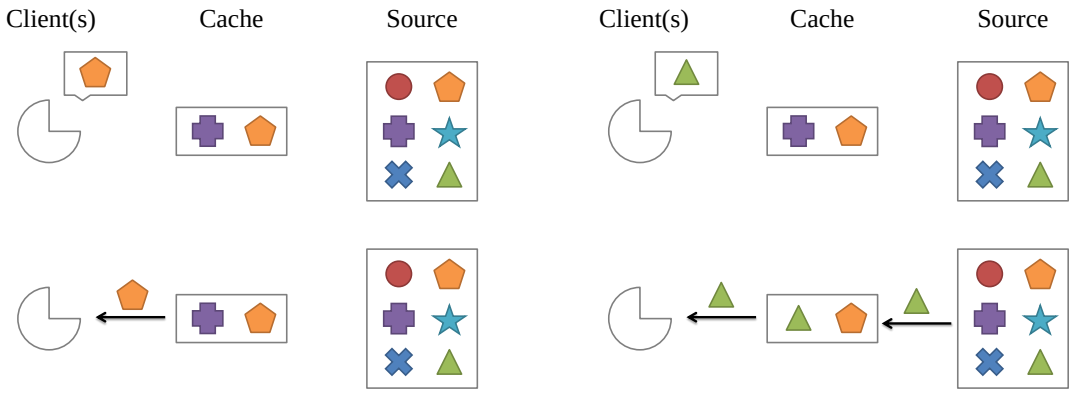
Caching strategies are implemented on systems where one or more *clients* request data from one or more *sources*, with a broad definition of *client* and *source*. The *cache* component acts as an intermediary between the source and the client, intercepting all client requests. At each client request, the cache verifies if it has already stored the requested data. If it is the case, the request is a *hit* (Fig. 1.1a) on the cache and the data is transferred to the client directly from it. Otherwise, the request is a *miss* (Fig. 1.1b) and the cache forwards the request to the source, possibly saving the requested data at the expense of another. The rule used for deciding when and what to erase and store is called *cache policy*.

Caching provides an increased system efficiency thanks to the so-called *locality principle*. This principle can be informally defined as the tendency for clients to request only a small subset of the source's data at any given time frame. Among the factors [60] affecting locality we can quote:

- Content popularity
- Temporal correlations of requests
- Spatial correlations of requests

The terms *temporal locality* and *spatial locality* are usually interchangeably used with temporal and spatial correlation.

A good caching strategy will exploit these tendencies by implementing an eviction policy that maintains most of the "local" subset most of the time. Due to the universality of the locality principle [18],



(a) *Cache handling a hit request* (b) *Cache handling a miss request*

Figure 1.1: *Generic caching scheme*

caching strategies have been successfully used in many contexts. We now proceed to review some of them.

Some Classical Caching Applications

Some well known examples of caching strategies can be found in Table 1.1.

Cache System	Data	Client(s)	Source(s)
CPU	Memory Pages, Instructions	CPU	RAM
DNS	Name-IP, Name-Name Pairs	DNS Servers	DNS Servers
Web (Client)	Multimedia, HTML Files	Web Users	Web Servers
Web (Server)	Multimedia, HTML Files	Web Users	Web Servers, Repositories

Table 1.1: *Some applications of caching strategies*

The most classical use of caching is found within the context of computer architecture. A CPU cache is an intermediate memory between the CPU and the RAM, with a smaller capacity and faster access time, that stores memory pages and instructions. In this application, temporal correlations (repeated instructions for loops) and spatial correlations (continuous blocks of referred memory) are exploited. Without CPU caches, either the performance would be degraded, since RAM access would not be at the level of modern CPU clock speed; or it would be costly to avoid the latter bottleneck, since it would then be necessary to have large quantities of expensive fast access RAM. For a comprehensive introduction to this subject, see [52, Ch. 5.3].

In the context of network architecture, caching is essential for the *Domain Name System* (DNS),

which is a distributed database in charge of translating domain names (e.g. `www.orange.com`) into IP addresses (e.g. `185.63.192.20`). In this system, every DNS server in the network acts as a cache for hostname-IP pairs. Again temporal and spatial (regional) correlations are exploited, but in addition, popularity is also exploited since a few hosts take most of the queries [39]. The latter is critical, since on DNS a query is recursively repeated in each server until one provides the answer back. Caching allows the answers to be distributed in the system according to the demand and fulfills the high responsiveness requirements of the system. For details about DNS, see [41, Ch. 2.5].

Another application related to networks is the caching of Web resources such as HTML pages, images and video. In this domain, caching is applied at two levels: The first one is at application level, wherein a cache is simply a collection of files in the user's disk; the second one is at the network level wherein a cache is a proxy server placed near the final users.

While application level caches are widely used in web browsers and smartphone apps, they can only exploit temporal correlations of a single user. On the other hand, cache servers increase more the network efficiency since they also exploit the spatial correlation of all users in the region where servers are placed and popularity of content as well. This system have been in use by companies and universities since the early days of the web, but its efficiency has declined in recent years due the rise of encrypted communications and the refusal of content providers to replicate their data without authorization.

Nonetheless, caching still plays an important role in today's Internet. In fact, caches are a key component in *Content Delivery Networks* (CDNs) and the *Information Centric Networking* (ICN) paradigm. These architectures address the problems arising from the mismatch between the original and current purpose of the Internet: while originally conceived to access remote resources in a host-to-host fashion, today's Internet is mostly used for content retrieval. We briefly review these architectures in the following.

Caching in the Internet of today and tomorrow: CDNs and ICN

CDNs are large networks of servers whose objective is to efficiently distribute content to Internet users. To accomplish this, CDNs are geographically dispersed so that every user is close to an *edge server* of the network (see Figure 1.2) which performs both replication and caching. In consequence, the CDN architecture exploits all three locality factors to efficiently deliver content to users. Additionally, they are transparent from the user's point of view: their content requests automatically re-routed by the CDN to the most appropriate edge server to deliver it.

Initially conceived as a load balancing strategy to cope with "flash crowd" problems, CDNs have now become crucial for the functioning of the Internet. Among the factors that helped this phenomenon to occur are the huge number of users on today's Internet, the increasing availability of high-speed broadband access, and the rise of web video and User Generated Content (UGC). As an example, more

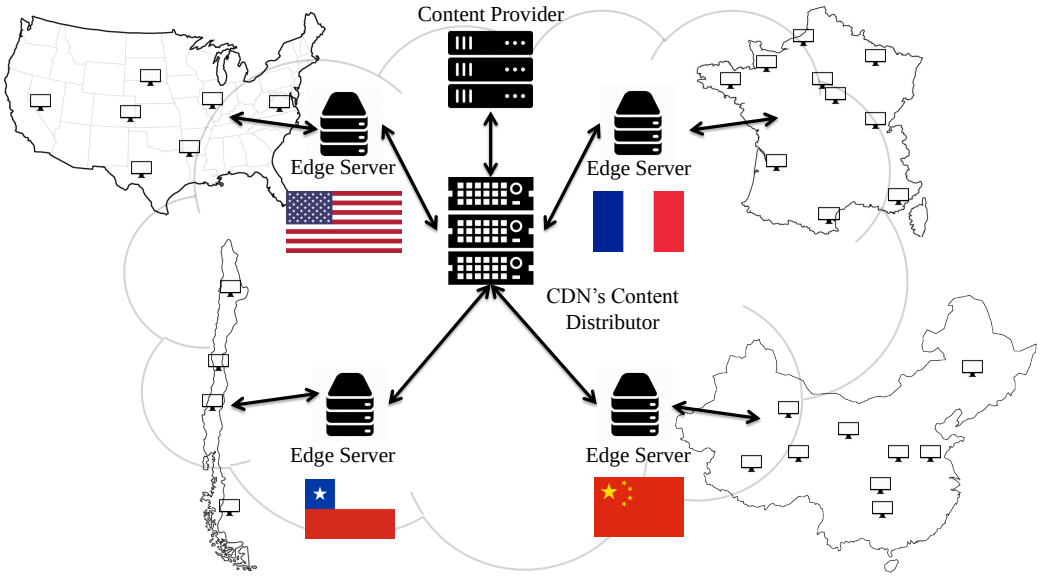


Figure 1.2: *A simplified CDN model (based on [8])*

than 60% of video traffic is nowadays delivered by CDNs and it is expected that this number will increase to more than 70% by 2019 [11]. Both large content providers such as Facebook or YouTube, and ISPs use CDNs to deploy their services. They either implement their own CDNs or contract specialized companies such as Akamai Technologies and Limelight Networks.

CDNs are beneficial to all actors involved in the process. On the one hand, content providers and users benefit from high availability, low access times and an overall increase in Quality of Service. On the other hand, ISPs obtain a decreased usage of their network resources and increased robustness to traffic bursts since CDNs also act as load balancers. For a comprehensive treatment on the subject of CDNs, see [8, 63].

While the CDN architecture has been an organic response to the changes in the Internet's utilization, the ICN architectures address this problem by redesigning the network layer. In particular, one of its main objectives is to have an scalable network architecture by adding content delivery capabilities to the network layer. To this aim, the principal design choice in ICN architectures is to decouple the naming from the location of content. Thus, such an architecture, the network is able to deliver content by just knowing which specific data the client is requesting (see Fig. 1.3). In practice, this design principle is implemented by replacing named hosts with named data, which allows to implement the desired scalability techniques at the network layer.

In particular, all major ICN proposals feature in-network caching, allowing data to be cached in routers and other intermediary devices. Many aspects of caching in ICN architectures are the object of active research nowadays. In particular, the interplay between caching and routing policies require

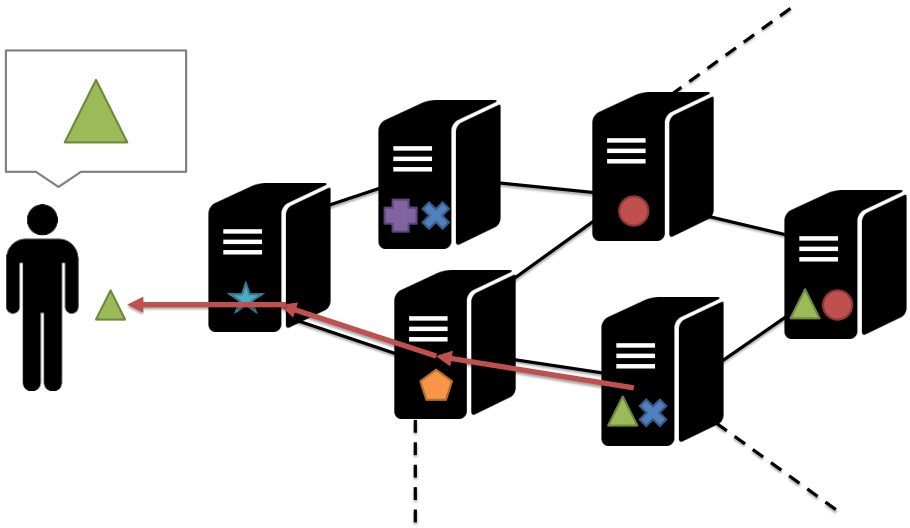


Figure 1.3: *Simplified ICN retrieval model*

the understanding of their interactions in such an architecture. For an overview of the different ICN architectures, advantages and challenges, see Alghren et al. [4].

The ICN and CDN architectures have renewed the interest in cache performance modeling in recent years. In consequence there has been an increased demand for dimensioning and exploratory tools to analyze the various "what-if" scenarios arising in many studies on these subjects. We continue this introductory exposition with an overview of the performance evaluation process.

1.2 Performance Evaluation

In simple terms, *performance evaluation* is the process of estimating the *performance indicators* of a system as a function of its characteristics. A performance indicator measures "how well" the system functions in an specific area of its behavior: for example, in computer networks, usual performance indicators include its throughput, loss rate and latency. Performance evaluation is an important task for the dimensioning of systems, in which the key question to answer is "What is the minimum amount of resources required for the system to attain a given performance level?".

Among the available performance evaluation methods, we concentrate our efforts on the *modeling* approach rather than simulation or experimentation. In this method, the system is abstracted to a mathematical model and its characteristics to the model parameters. The model is then analyzed to obtain estimates for performance indicators in terms of the input parameters. This method has two advantages with respect the others:

- First, performance formulas provide intuition and insight of how the system works in function of the parameters. This intuition takes more time and resources to build the other methods;
- secondly, calculating formulas in a computer is less resource-demanding than simulation or experimentation, allowing to obtain results faster. This is convenient when we want to analyze and compare many "what-if" scenarios.

This process of obtaining estimates from mathematical models is what is often called *performance evaluation*. In our work, however, we also consider the procedures to relate traffic measurements with the model. In the following, we discuss these data related processes and how, in conjunction with the mathematical analysis, they conform a workflow for performance evaluation.

Data Driven Performance Evaluation

In Figure 1.4 we show our workflow for performance modeling.

The workflow starts with a *model definition* procedure, in which we search the key properties of the system to be accounted for the model. We determine the importance of a property by measuring its impact in the performance indicators to estimate. The objective of this initial step is to obtain a model that is both mathematically tractable and accurate. Tractability is usually correlated with simplicity, and we thus prefer models with few parameters.

Once we have defined the model, we can proceed with the *mathematical analysis* (usually known as *performance evaluation*) to obtain theoretical estimations for the performance indicators of interest. These results can be readily used to explore the system performance by tinkering with the parameters. However, when evaluating a real system we must first pass by a *parameter fitting* procedure, in which we translate the system data into the model parameters. Finally, the fitted parameters joint with the theoretical formulas allow us to obtain the desired performance indicator estimate.

In the context of this workflow we now specify the setting and the questions we answer in this work.

Caching Performance Modeling

The performance of a cache server can be evaluated by various metrics. In our modeling, we focus on the *hit probability* or *hit ratio*, which is defined the portion of requests served by the cache, that is

$$\frac{\text{Number of hit requests}}{\text{Number of requests}}.$$

At this stage we make various hypotheses about the cache system. Note that we do not consider these assumptions as part of the "model definition" step in the above workflow.

We thus suppose that in the cache system:

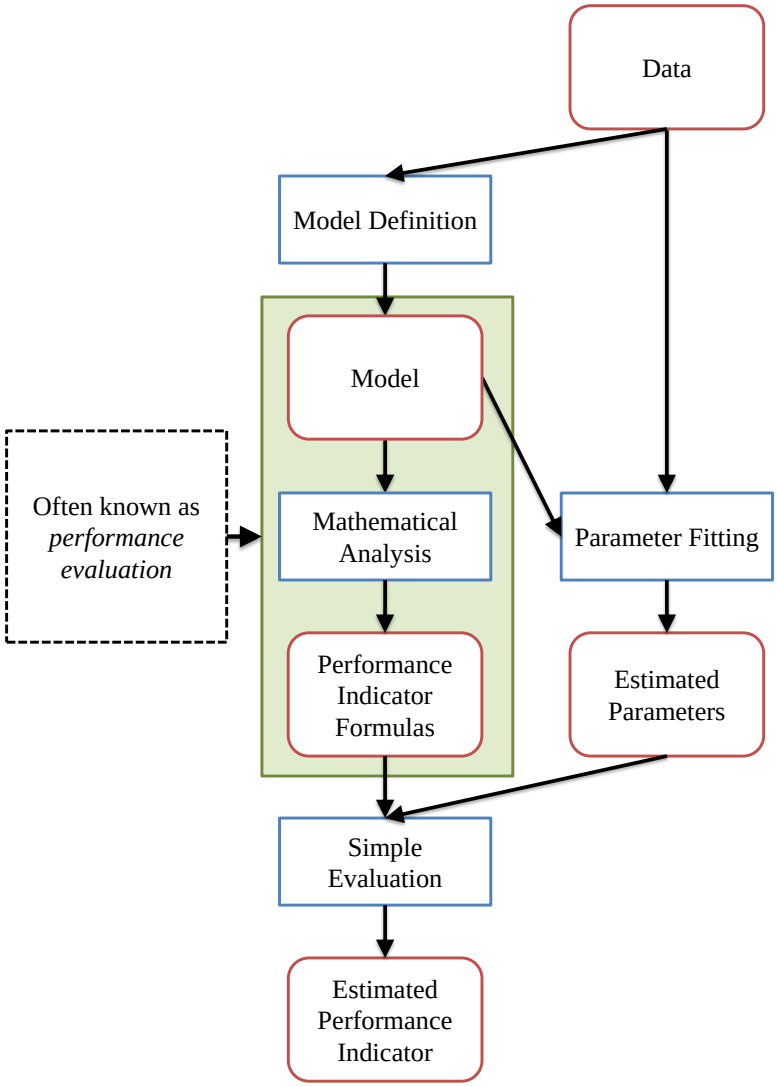


Figure 1.4: *Performance Evaluation Workflow*

- *Request are treated instantaneously*: a non-instantaneous treatment could affect the hit ratio if many requests to the same content arrive in bursts. However, we consider this effect negligible as it affects few request and thus it does not have a significant impact on the hit probability.
- *Disk access is instantaneous*: This affects other performance measures such as the latency. In consequence it does not affect the hit probability.
- *Network access is instantaneous*: Again this could affect burst of requests for the same content and the latency as well. We neglect this by simplicity.

- *All files have the same size*: We make this assumption since we do not have data to obtain the file size distribution. Note however that if this data is available, it can be incorporated to our model proposal (see the conclusion of Chapter 3).
- *The cache is always consistent*: When content is dynamic, a document can change in the source and thus it can differ from a cached version of it. In other words the cached content is *inconsistent* with the source. We do not consider this issue by simplicity.

Under these hypotheses, we model a cache as a set with $C \geq 0$ elements. We call the quantity C the *cache size* or *capacity*. The hit probability is then a function of the capacity $C \mapsto q_C$ expressed as

$$q_C = \frac{\text{Number of hit requests in a cache of size } C}{\text{Number of requests}}.$$

The set of stored elements evolves according to the caching policy, which is defined by a storing decision $\mathcal{D}$ and a replacement algorithm $\mathcal{D}$ [56]. Upon a miss event, the cache applies $\mathcal{D}$ and, if positive, applies $\mathcal{R}$ to make room for a new document. Some common caching policies are:

LRU	$\mathcal{D}$: Always	$\mathcal{R}$: Least recently referenced document
LFU	$\mathcal{D}$: Always.	$\mathcal{R}$: Least frequently referenced document
q-LRU	$\mathcal{D}$: With probability q	$\mathcal{R}$: Least recently referenced document
FIFO	$\mathcal{D}$: Always	$\mathcal{R}$: Last in "first in first out" queue
RANDOM	$\mathcal{D}$: Always	$\mathcal{R}$: Randomly chosen
TTL	$\mathcal{D}$: Always	$\mathcal{R}$: After a document's timeout (independent of miss).

From this list, LRU is the policy that has received the most attention. To implement an LRU cache in our model we make the set representing the cache a self-organizing list by adding the following update mechanism (see Fig. 1.5): Upon a document request:

- If the document is already stored in the cache, then it is moved to the front of the list, while all documents that were in front of it are shifted down by one slot;
- otherwise, a copy of the requested document is placed at the front of the list, and all other documents are shifted down by one slot, except the last document which is eliminated.

In this work, we concentrate in this policy since it is simple to implement, has proved to be efficient and does not need any parameter tweaking. All our modeling assumptions plus the fixation of the cache policy makes the function $C \mapsto q_C$ depending only on the stochastic model chosen for the document request sequence. Now that we have set up the cache model, we ask below three relevant questions regarding each of the key steps in our workflow.

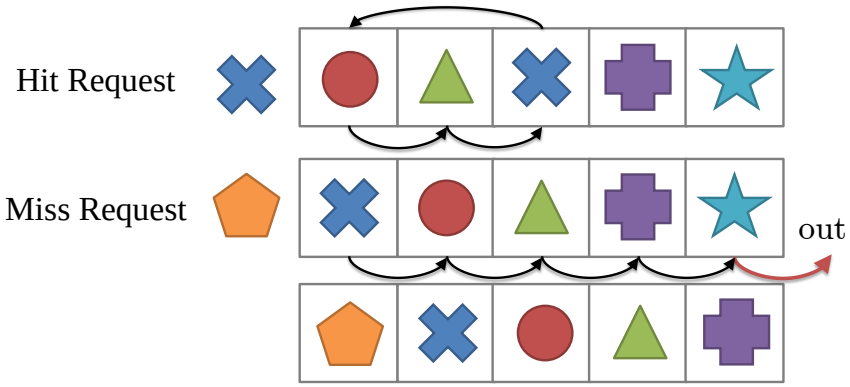


Figure 1.5: *The LRU policy handling a hit and a miss request on a cache of size* $C = 5$.

Model Definition:
• Which assumptions can be reasonably made on the stochastic process modeling the request sequence?
Mathematical Analysis:
• Can we deduce a theoretical expression or approximation for $H(C)$ from such a model?
Parameter Fitting:
• How can we reliably estimate the parameters of such a model from traffic traces?

The aim of this thesis is to contribute to answer these questions. Before stating our contributions we review the related work regarding each of the latter questions.

1.3 Related Work

We review the related work for each of the non trivial steps of our workflow (Figure 1.4), namely model definition, mathematical analysis and parameter fitting, in the setting of LRU cache performance evaluation.

Model Definition

To define traffic model from traces, we use the so-called *semi-experimental method*. A semi-experiment consist in a randomization of the traffic trace designed to break a correlation structure on it, and an *oracle* to determine if the randomized trace is similar to the original traffic trace. Thus, it allows us to decide whether the broken structure is relevant. The oracle depends on the problem at hand, but is usually the difference of a relevant quantity calculated numerically or via simulations.

This procedure was first proposed by Erramilli et al. [20] to study long range dependency (LRD)

on Internet traffic. With the of studying LRD in Internet traffic, Hohn et al. [33] generalized this procedure and coined the term "semi-experimental". In the context of caching performance evaluation, this methodology has been used by Traverso et al. [59] to propose a shot-noise point process model for the request sequence. Other applications include the study of wireless traffic [54] and bandwidth estimation for a model based on Kelly networks [6].

Stochastic Models and Theoretical Hit Ratio Estimates

Independent Reference Model

From the early research on LRU cache performance evaluation, the main stochastic model for analysis has been the *Independent Reference Model* (IRM). The main feature of IRM is its simplicity: the requests are modeled as an i.i.d. sequence taking values on the finite set $\{1, \ldots, K\}$ called the *catalog*. Content popularity is then modeled by the distribution of the sequence: $\mathbb{P}[X = k] = \lambda_k$ for $k = 1, \ldots, K$.

Many of the key ideas used in this thesis have been already explored in the f ramewokr of IRM. In some cases, LRU caches were not directly studied, but rather the related Move-to-Front (MTF) search list of size K. These systems are closely related due a relationship between LRU cache misses and the MTF list search cost. In fact, we have the event equality

$$\{\text{Miss in an LRU cache of size } C\} = \{\text{Search cost in an MTF list is larger than } C\}\,. \tag{1.1}$$

Under IRM, the dynamics in the cache is ergodic and thus the stationary miss probability is the main quantity of interest. Notably, Flajolet et al. [22] derived exact combinatorial formulas for the miss probability for an arbitrary popularity.

The Che Approximation

Apart from some special cases, the latter formulas derived by Flajolet et al. contain sums with an exponential number of terms, which renders them practically intractable. A key contribution towards approximative approaches was made by Fill and Holst [21] who re-derived the results already obtained by Flajolet et al. by embedding the IRM sequence into a Poisson process. In this model, each document has its own request process ξ_k, which is a marked Poisson process with request rate λ_k and fixed mark k. The request sequence is then modeled with the superposition of all document processes. There is an ambiguity in the literature because the latter embedded process and IRM are often conflated.

The above embedding technique enabled Che et al. [10] to propose a heuristic method, now called the *Che approximation*, to calculate hit probability for a LRU cache. In their work, they express the hit probability of a LRU cache in terms of a family of exit times $\{T_C^k\}_{i=1}^K$ that represent the elapsed time between the document request and its eviction. Specifically, since an object at the top of the cache is

evicted after C documents were requested, we have that T_C^k is the following first passage time

$$T_C^k = \inf\left\{ t > 0 : \left(\sum_{j=1;\, j\neq k} \mathbb{1}\{\xi_j(t) \geq 1\} \right) \geq C \right\}.$$

In order to simplify the analysis, the authors argued that in the case of a Zipf popularity distribution, we can approximate the hit probability as follows:

Che.1 First, we assume that the whole family has the same distribution, that is, $T_C^k \overset{\mathrm{d}}{=} T_C$ for some random time T_C.

Che.2 Secondly, we assume the time T_C can be well approximated by a single constant t_C called the *characteristic time*. This time is defined as the solution to the equation:

$$C = \sum_{k=1}^{K} \left(1 - e^{-\lambda_k t_C}\right). \tag{1.2}$$

Che.3 Finally, the hit probability is then estimated as the average

$$q_C = \frac{\sum_{k=1}^{K} \lambda_k \left(1 - e^{-\lambda_k t_C}\right)}{\sum_{k=1}^{K} \lambda_k} \tag{1.3}$$

The Che approximation proved to be empirically accurate even for non Zipfian popularity profiles, so much that it is now the de facto method to estimate the hit ratio for a LRU cache [26, 7, 28, 24, 30, 55].

In parallel, another approximative approach was proposed by Jelenković [35], who studied an asymptotic equivalence for the miss probability as K and C grow large. Later, using some of the latter results with the Poisson version of IRM, Jelenković and Kang [36] developed an asymptotic estimation for the miss probability in the case where the popularities have a Zipf profile. They use the scaling $C = \delta K$ with fixed $\delta < 1$ to obtain the asymptotic estimation for the miss probability. Although they do not mention it, their argument is in fact a rigorous justification of the Che approximation.

The question of quantifying the error incurred by the Che approximation has been partially answered by Fricker et al. [26], where the authors provide a justification for a Zipf popularity distribution when the cache size C grows to infinity and scales linearly with the catalog size K. The error incurred by the approximation is estimated for the exit times but not, however, for the hit probability.

Beyond IRM

While the IRM framework is tractable, its source of locality is only the popularity distribution. In fact, the very definition of IRM immediately implies that there are no temporal correlations in the request sequence.

Since web traffic evidences the presence of temporal correlations [38], there has been interest in models that take into account such phenomena. A Markovian model was proposed by Psounis et al. [53] and later analyzed by Panagakis et al. [51] who obtained approximate expressions for the miss probability by adapting the Che heuristic. In a similar vein, Jelenković and Radonavić [37] propose a Semi-Markov stochastic process and estimate the asymptotic miss probability as well.

More recently, Traverso et al. [59] proposed a shot noise point process as a traffic model for which Garetto et al. [29] obtained theoretical estimates for the miss probability by adapting the Che heuristic. The approximation's accuracy was treated by Leonardi and Torrisi [42], where limit theorems for the exit time are provided for C going to infinity, together with an upper bound of the error on the hit probability.

Parameter Fitting

Content Popularity Estimation

Content popularity is the main parameter of the IRM model. Due to the fact that popularity distributions usually exhibit a power law behavior, a common method to estimate them is to fit its rank-frequency distribution in double logarithmic scale. This approach has been recently criticized by Clauset et al. [12]. The main issue is that the rank-frequency plot is not a reliable statistic since, for example, it can exhibit power-law behavior even if the ground-truth does not.

Despite these problems, the use of the latter method is still pervasive in performance evaluation in the case of IRM [27, 30] and traffic characterization studies [31, 34, 9]. Authors try to improve these methods by means of various adjustments. In [34], for example, authors separate in three parts the rank-frequency plot and adjusting different curves in each piece, and in [31], authors adjust "stretched exponential" curves instead of power-laws.

The latter adjustments indeed solve some of the fitting issues. However, in previous studies [30], it has been noted another problem in the context of performance models, which arises from the fact that, within the model, objects can have zero requests. In consequence, from the point of view of the network operator, objects with no request are not observed in traces. In statistical jargon, the sample is *zero-censored* and not taking this fact into account leads one to underestimate the catalog size, which has an impact on the conclusions drawn from the fitted model.

Maximum Likelihood Estimation

In the present work, we address the issues presented above by using Maximum Likelihood (ML) estimates. This method allows us to seamlessly handle the zero-censored case and it is proposed by Clauset et al. [12] as a robust method to fit heavy tailed data, which is a common property in popularity distributions. Maximum likelihood methods have already been in use for flow size estimation [44] and call center modeling [50]. The latter work uses an approach similar to ours, but it is limited to a specific parametric model for non-censored data.

The statistical basis of our methods is the estimation of mixed discrete distributions, a subject that has been extensively studied in the literature. The non-parametric case has been addressed from two points of view: the first one searches the mixing density in the space generated by Laguerre polynomials with an exponential cut-off; the estimator is then obtained by a projection on the latter space [57, 13]. However, this estimation method converges slowly with the sample size unless the density belongs to the aforementioned space. We therefore base our methodology on the second point of view, which assumes the mixing distribution to be a sum of Dirac masses. The estimation methods are then similar to an Expectation-Maximization scheme (EM) [43]. As regards the parametric case, EM schemes for finding the parameters of the mixing distribution are provided for many families in [40]. In both parametric and non-parametric cases, the estimation algorithms do not handle the case of censored data, and thus we have to rely on an all-purpose nonlinear optimization solver to obtain our results.

1.4 Contributions and Organization

We here briefly describe the contributions of this thesis at each stage of the performance evaluation workflow (see Figure 1.6). In addition to the latter contributions, we present in Chapter 5 a global conclusion and perspectives to the works we have developed. Chapter 6 is an appendix containing some technical proofs and a brief description of the key algorithms used in this work.

Chapter 2: Model Definition

In this chapter, we propose a parsimonious traffic model which allows us to accurately estimate the hit probability. With this aim, we study the correlation structure in two large traffic traces by means of the semi-experimental method. Three semi-experiments were performed on the traces, each one being a randomization procedure targeting a specific correlation structure in the requests. In each experience, we compared the hit probability of the original sequence to that of the randomized sequence.

The main conclusions we obtained from these experiences are the following:

- At short timescales, in the order of minutes, the influence of dynamic of the catalog popularity is negligible and thus IRM is still a good model in this case;

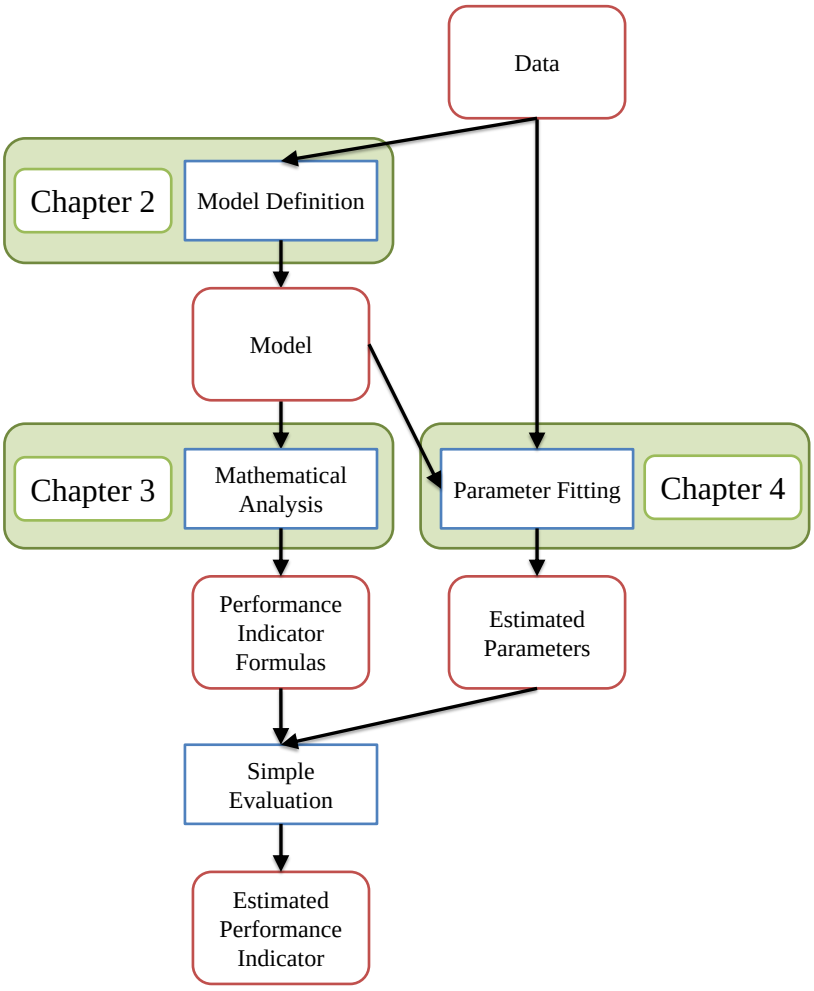


Figure 1.6: *Contributions in the Performance Evaluation Workflow*

- at longer timescales, this influence is significant on the hit probability;
- at long timescales, the request sequence for a given document can be modeled as an homogeneous Poisson process within a finite lifespan and the times in which documents become available can be modeled as an homogeneous Poisson process. These requirements are captured by the well known *Poisson cluster process*.

The above contributions were published in the first part of the article presented at the ITC26 conference [3] in collaboration with Bruno Kauffmann, Alain Simonian and Yannick Carlinet. The contributions presented here were discovered simultaneously and independently from those Traverso et al. [59] which propose a model similar to ours. However, our proposal differentiates in that we consider

a random bivariate model for the source of catalog dynamicity.

Chapter 3: Theoretical Hit Ratio Estimation

In this chapter, we mathematically analyze a generalization of the model we proposed in Chapter 2 in which the document request sequence is an inhomogeneous Poisson processes. The aim of the analysis is to rigorously obtain an estimation on the hit probability for large cache size. The main contributions are:

- We use the Palm distribution associated with the process to set up a probability space in which a document can be analyzed independently from the rest;
- the above stochastic setting allows us to derive exact integral formulas for the expected number of misses;
- We obtain an asymptotic expansion for the expected number of misses by means of a scaling of meaningful quantities of the system. This expansion justifies the use of the Che Approximation for this traffic model;
- We validate the theoretical results by comparing them to the hit probability obtained via simulations.

A first version of these contributions is provided in a simple case in the above mentioned conference article [3], and later generalized and refined in [1] in collaboration with Carl Graham and Alain Simonian. The main differences with regards to similar works are the use of Palm theory to rigorously set up the analysis from beginning to the end; and secondly, our estimates are explicitly calculated in terms of the system parameters, whereas in the previous results the latter bound depends on an additional variable, whose optimal value is not explicitly given.

Additionally, the lemmas we developed in this work enabled us to obtain expressions for the transient hit probability for the IRM and the IRM-Mixture models used in Chapter 4.

Chapter 4: Parameter Fitting

In this section we tackle the problem of parameter fitting for a modified version of the IRM model. In this model, that we call IRM-Mixture (IRM-M), popularities are treated as a random sample from a fixed distribution instead of being fixed. IRM-M model the same localities as IRM, but in addition, it has the advantage of being tractable for parameter estimation. Additionally, IRM-M can be seen as an intermediary model between IRM and the cluster model we propose in Chapter 2, and thus our contributions to the inference problem for IRM-M can be helpful in the development of a method for cluster traffic models. The main contributions are the following:

- We propose a Maximum Likelihood (ML) method to estimate the popularity distribution from data traces;
- we show that the latter method can seamlessly handle the fact that documents with zero requests are not observed.

The contributions of this section were presented in the VALUETOOLS 2015 conference [2] in collaboration with Bruno Kauffmann.

Contributed Articles

[1] F. Olmos, C. Graham, and A. Simonian. Cache Miss Estimation for Non-Stationary Request Processes. arXiv:1511.07392, 2015. Submitted.

[2] F. Olmos and B. Kauffman. An Inverse Problem Approach for Content Popularity Estimation. In *9th International Conference on Performance Evaluation Methodologies and Tools (VALUETOOLS)*. EAI, 2015.

[3] F. Olmos, B. Kauffmann, A. Simonian, and Y. Carlinet. Catalog Dynamics: Impact of Content Publishing and Perishing on the Performance of a LRU cache. In *26th International Teletraffic Congress (ITC)*. IEEE, 2014.

Chapter 2

Model Definition

The objective of this chapter is to define a traffic model to accurately estimate the performance of a LRU cache. As noted in the introduction, web traffic contains temporal correlations which are not accounted for by simple models such as IRM.

We thus aim at obtaining a model that reflects these correlations while remaining simple enough to be mathematically tractable. To accomplish this, we apply the semi-experimental method to two large traffic traces and identify the key structural properties of the request sequence relevant to a LRU cache. These characteristics give us clues about the assumptions we can safely make when modeling the traffic. We eventually propose a model based on cluster point process, and validate it by comparing its hit probability predictions with the empirical findings.

To start this chapter, we describe the origin and treatment of the traffic traces that form the basis of our work.

2.1 Datasets

We have gathered two datasets from two services, which have different traffic profiles. See Table 2.1 for a summary.

The first dataset, hereafter named `#yt`, captures YouTube traffic of Orange customers located in Tunisia. We have access to the logs of a transparent caching system set up in order to offload the country international connection. This system is a commercial product from a large company specialized in the design and management of CDNs.

Operational constraints, such as the limited disk space available for the logs on the cache system, made the latter system to miss requests when the traffic load was at its peak. We could estimate, however,

	#vod	#yt
Origin	France	Tunisia
Traffic Type	Video-on-Demand	YouTube
Period	2008 – 2011	January – March 2012
Raw Requests	3.4 Million	420 Million
After Treatment Requests	1.8 Million	46 Million
Total Documents Requested	120 Thousand	6.3 Million

Table 2.1: *Summary of datasets*

that the number of missing requests was less than 1% in January and February 2012, and less than 9% in March. Since the number of missed requests increases afterwards, we will focus our study only on the period from January to March 2012.

In this observation period, we collected around $420\,000\,000$ requests from about $40\,000$ IP addresses to $120\,000\,000$ video *chunks*. For each chunk request in this trace, the logs contain the user (anonymous) IP address, a video identifier, the time-stamp of the end of session, the number of transmitted bytes, the duration of the HTTP connection and the beginning and ending position of the specific chunk requested, the latter information being available for 96% of the data.

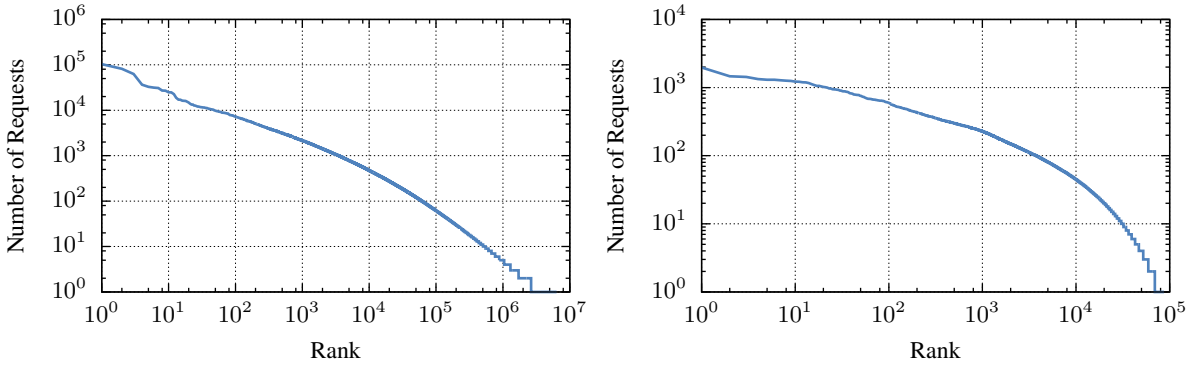
The second dataset, hereafter called `#vod`, comes from the Orange Video-on-Demand service in France. This service proposes to Orange customers both free catch-up TV programs, pay-per-view films and series episodes. Probes deployed at the access of the service platforms recorded video requests from June 2008 to November 2011. The data amounts to more than $3\,400\,000$ requests from $60\,000$ users to $120\,000$ videos. The records in this trace consist in the request timestamp, an internal client (anonymous) identifier and a video identifier.

Treatment

Recall that we wish to base our modeling on document requests. We must first treat the `#yt` trace by consolidating the chunk request sequences so as to recover the document sessions that generated them.

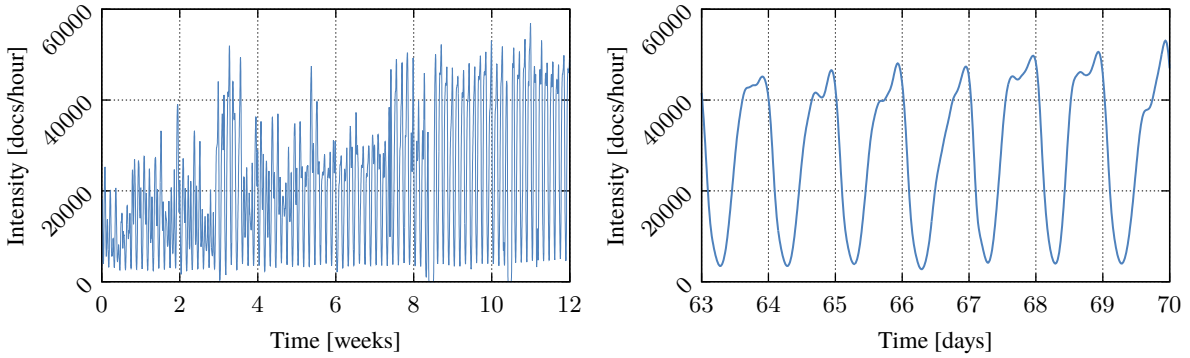
We achieve the latter by using the available chunk information, namely the beginning and ending position of the chunk within its file. This information allows us to chain consecutive requests to the same document with adjacent chunk positions, and aggregate them to a single request. However, as stated before, the chunk information is not present in around 4% of the entries. To aggregate the requests in this subset, we use a time-window criterion: we conflate all requests made by the same user for the video that have inter-arrival time smaller than 8 minutes. This threshold corresponds to the 95% percentile of the length session distribution of requests with chunk data[1] The result of these procedures is our working `#yt` dataset consisting of more than $46\,000\,000$ requests to about $6\,300\,000$ unique documents.

[1]We select this percentile to exclude sessions of extreme length (there were 185 lasting several days)



(a) *#yt trace* (b) *#vod trace*

Figure 2.1: *Number of requests*



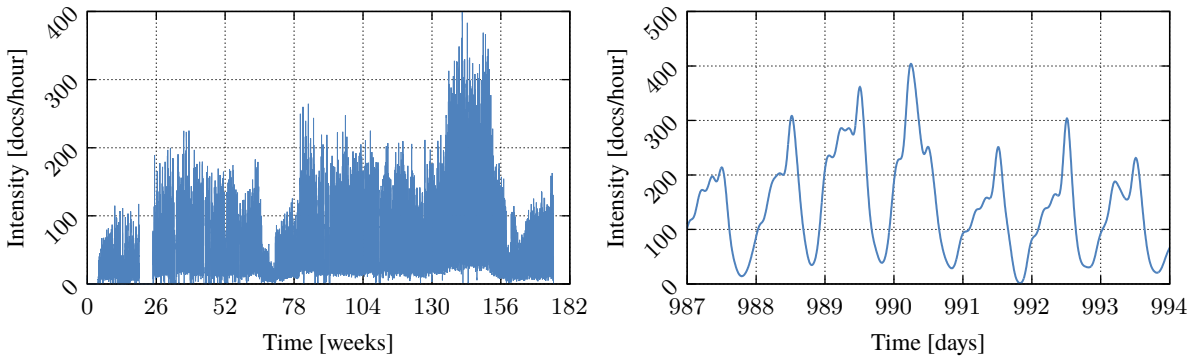
(a) *Full range* (b) *Zoom at the 9th week*

Figure 2.2: *Request intensity for #yt*

In Figure 2.1a, we show the number of requests of a document as a function of its rank in a log-log scale. We observe that this profile is not that exactly that of a Zipf law, since it is not a straight line. With regards to the intensity of requests (Fig. 2.2), there is a clear daily and weekly periodicity with a slight increase in the load over time.

In the case of the #vod trace, there was no need of the above consolidation procedure. However, the trace contained two types of "content surfing" entries: the first consisted in requests to movie trailers; the second consisted in requests with very short duration. We considered that these kinds of requests were not relevant in terms of cache performance, and we have therefore discarded them from the dataset. The working #vod dataset resulted in around 1 800 000 requests to more than 87 000 different objects.

In this case, the number of requests profiles is even more pronouncedly non-Zipfian (Fig. 2.1b) than in the #yt case. As for the request intensity (Fig. 2.3) we observe again a weekly periodicity but with



(a) *Full range* (b) *Zoom at the 141st week*

Figure 2.3: *Request intensity for* `#vod`

volatile load in each week.

2.2 Semi-Experiments

Intuitively, any time correlation in the request sequence has an impact on the performance of a LRU cache. On one extreme, for a sequence with maximal time correlation (all request are made to a single document), all requests except the first are hits. On the other extreme, a totally uncorrelated sequence (no document receives two requests) will obtain no hits.

We thus investigate how three correlations structures in our data can impact the hit probability of a LRU cache. Specifically, the three structures we investigate are:

(i) The correlations between all request times

(ii) The correlations between the document apparition times

(iii) The correlations between the request times within an individual document request sequence.

Additionally, in the case (i), we look for the time scale where such correlation structure starts to be significant.

For this endeavor, we use the *semi-experimental method* [33]. A semi-experiment consists in two procedures:

1. We shuffle the request sequence in a way that destroys the targeted correlation structure.
2. We use an *oracle* to tell if the original and shuffled sequences differ significantly.

Note that since we only shuffle the sequence, the number of requests per document remains the same

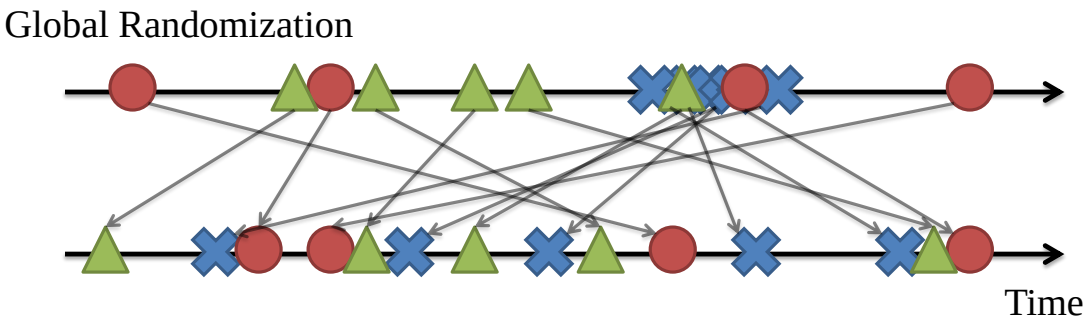


Figure 2.4: *A schematic view of the global randomization that shuffles all request times.*

after the procedure. Thus, our semi-experiments allow us to measure the impact of temporal correlations while leaving content popularity unchanged.

The choice of oracle is application-dependent: For example Hohn et al. [33] use the wavelet transform as oracle since they studied LRD properties of Internet traffic. In our case, the oracle consist in first simulating a LRU cache fed with the traces and then compare the resulting hit probability curves. If they differ significantly, we infer that the broken structure is relevant for the performance of a LRU cache. Our curve discrepancy measure is the *mean absolute percentage error* (MAPE): For a model sequence $(y_i)_{1\leq i\leq N}$ and empirical data $(x_i)_{1\leq i\leq N}$, the MAPE is defined as

$$\text{MAPE}(x,y) = \frac{1}{N}\sum_{i=1}^{N}\frac{|x_i - y_i|}{|x_i|}. \tag{2.1}$$

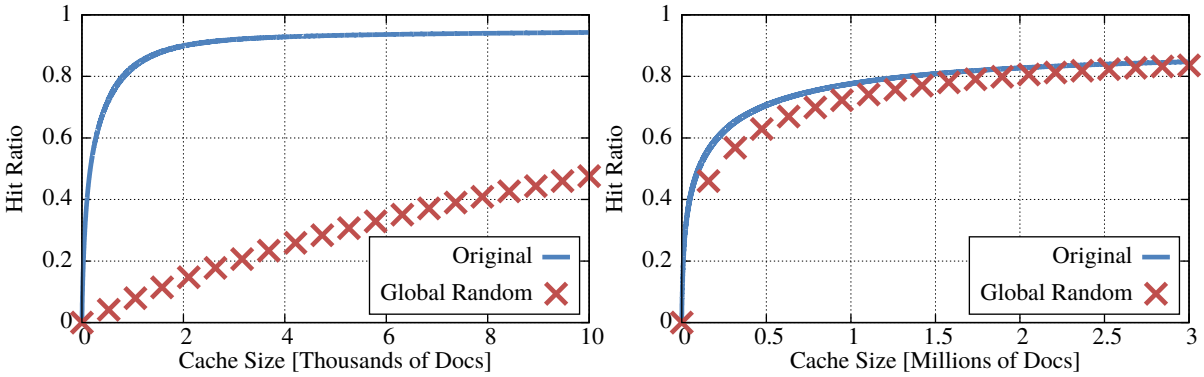
The MAPE will be our comparison measure for hit probability curves in the remaining of this work.

For the details of the key algorithms for simulation and trace generation see Appendix 6.2. We now proceed to explain in detail each semi-experiment and its findings.

Overall Correlation Between Requests

In this semi-experiment, we completely break the correlation structure of the request sequence by placing each request at an i.i.d. uniform time in the interval $[0, W]$, where W is the size of the observation window. Any trace shuffled in this manner leads to an IRM sequence, since the process destroys any dependence structure. Even more, conditional on the number of requests, the sequence for both individual documents and the ensemble is uniform after the shuffling. Thus, all request processes are Poisson and the resulting request sequence is an embedded IRM sequence. We call this procedure *global randomization* and show an example in Figure 2.4.

In Figures 2.5a and 2.5b, we compare the resulting hit ratio to that obtained with the original



(a) *#vod trace* (b) *#yt trace*

Figure 2.5: *Comparison of the hit probability of the original request sequence versus the results the global randomization.*

trace and observe that the hit probability of the latter is lower for any cache size in both datasets, but notoriously in the #vod case. Specifically, in the #yt case, the MAPE has a value of 5.0%; this value might seem low, but it comes mostly from the left of the curve. Since the left part of the curve is where practical cache sizes lie, this discrepancy, however low, is still important. As for the #vod trace, the MAPE amounts to 17.3% which confirms the huge difference observed above in Figure 2.5a. We thus conclude that, at this time scale, the correlation between requests is a meaningful factor for the performance of LRU caching and that the IRM assumption leads to an underestimation of the hit ratio, which can be very significant.

Correlation in Catalog Publications

We now examine how sensitive is our data with respect to the publication of new documents to the catalog. To this aim, we perform a *positional randomization*, which breaks the correlation structure between the first requests of documents, which we use as an estimate of the publication time. The procedure consists, for a given document, in leaving the inter-arrival times of its request sequence unchanged and jointly shift all of them by a random quantity, as shown in Figure 2.6. More precisely, let $\Theta_1, \Theta_2, \ldots, \Theta_k$ the request times for a document, then the randomization procedure is as follows:

- first, we draw a uniform random number U from the interval $[0, W - (\Theta_k - \Theta_1)]$;
- then we define the new request sequence $\Theta_1^*, \Theta_2^*, \ldots, \Theta_k^*$ by $\Theta_i^* = U + \Theta_i - \Theta_1$ for $1 \leq i \leq k$.

In both traces, the resulting hit probability shows no difference from the original, as observed in Figures 2.7a and 2.7b. The MAPEs in this semi-experiment are merely 0.3% in the #yt case and

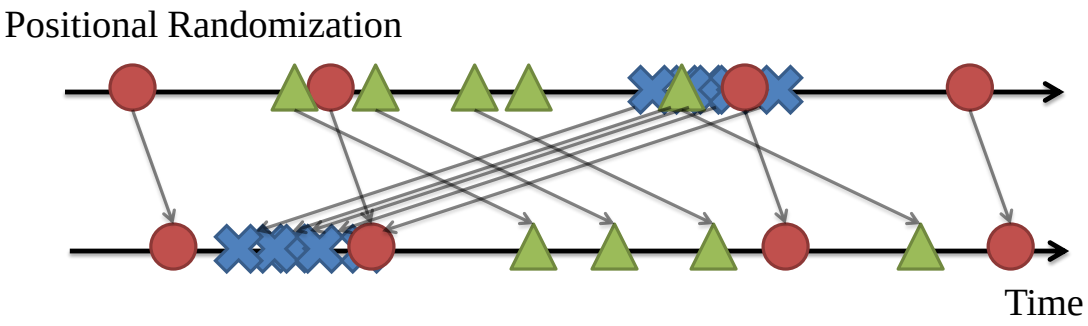
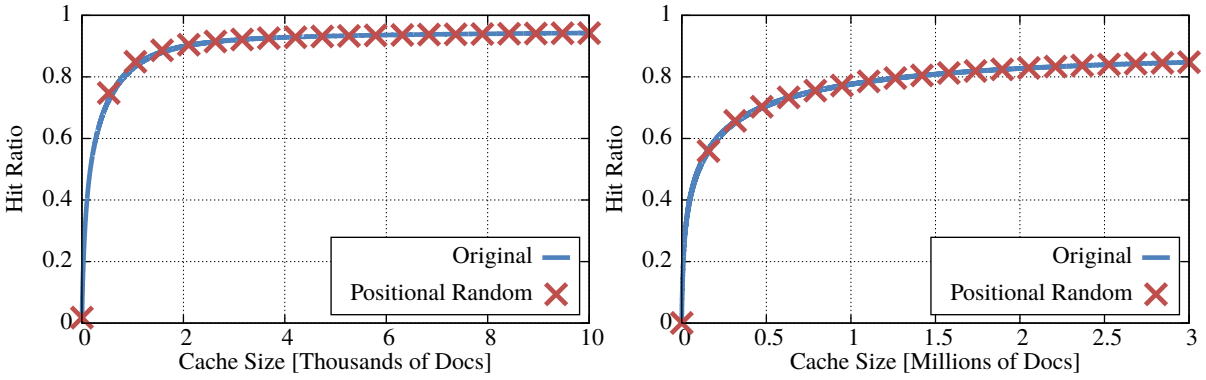


Figure 2.6: *A schematic view of the positional randomization that shifts the whole request sequence to a random location, preserving the order of inter-arrival times.*



(a) `#vod` *trace* (b) `#yt` *trace*

Figure 2.7: *Comparison of the hit probability of the original request sequence versus the results the positional randomization.*

0.1% in the `#vod` case. We therefore conclude that the correlation structure of catalog arrivals has no significant impact on LRU caching.

Correlation between Requests of a Document

In this semi-experiment, we aim at breaking the request dependence structure for each document. To achieve this, we perform a *local randomization* (Fig. 2.8): For a document with request times $(\Theta_k)_{k=1}^N$, we keep its first and the last request times fixed and only shuffle the ones in between at i.i.d. times following a Uniform $[\Theta_1, \Theta_N]$-distribution. Note that this procedure renders the request sequence of each document an homogeneous Poisson process within the interval $[\Theta_1, \Theta_N]$ and thus breaks any other correlation structure inherent to the request process of the document.

Figure 2.9a and 2.9b show that, although the resulting hit probability is slightly below the original

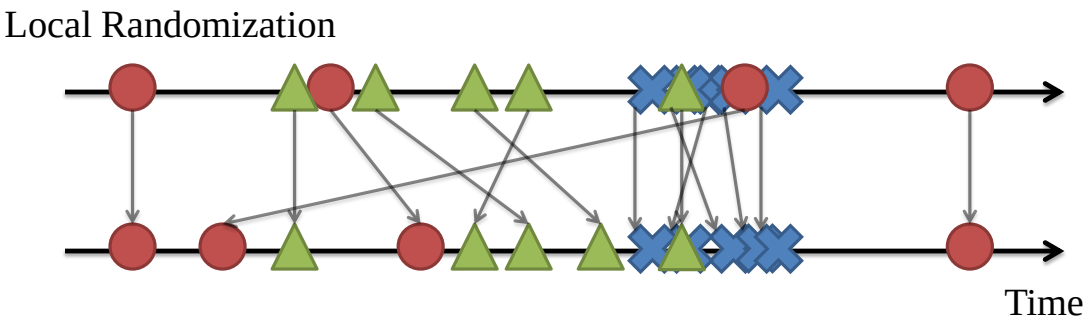
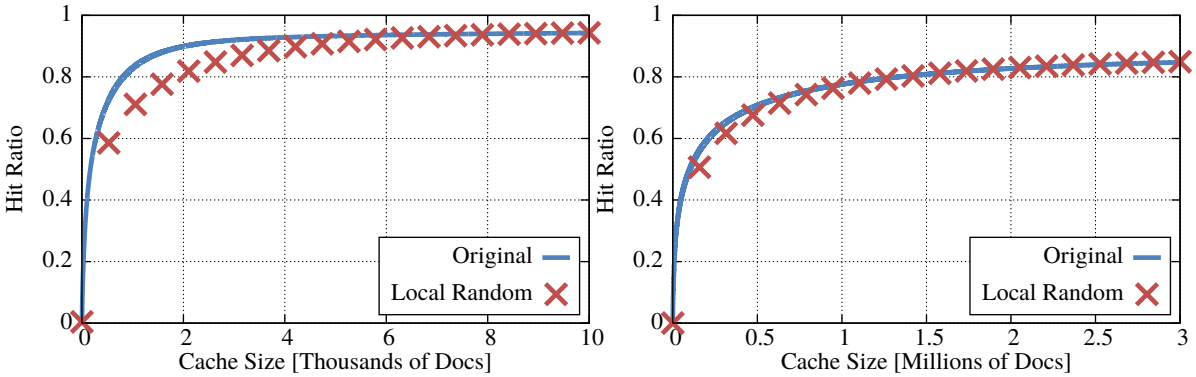


Figure 2.8: *A schematic view of the local randomization that fixes the first and last request and shuffles the times in the middle.*



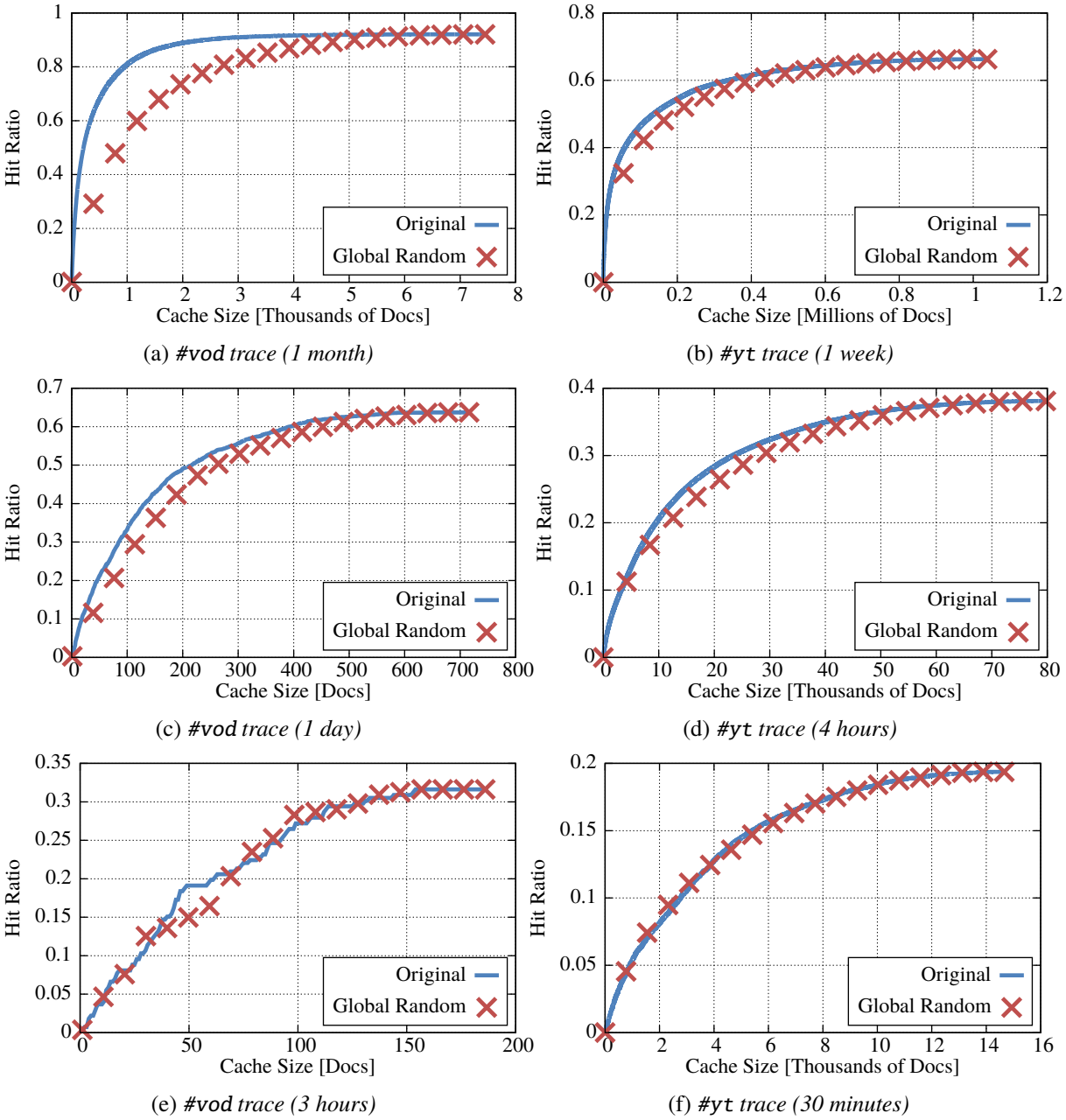
(a) *#vod trace* (b) *#yt trace*

Figure 2.9: *Comparison of the hit probability of the original request sequence versus the results the local randomization.*

for small cache sizes, the MAPE is just 1.6% in the `#yt` trace and 0.7% in the `#vod` trace. We thus conclude that the correlation among requests of a given document has little impact on LRU cache performance and we can safely neglect it for modeling purposes.

Relation between Correlations and Timescales

We now determine at which time scale the correlation between requests has an impact in the LRU performance. With this in mind, we design a slightly different semi-experiment where we first extract sub-traces of different time scales, choosing high load periods. Then we apply the global randomization semi-experiment to each of these shorter traces. For each dataset, we distinguish three time scales and the results for each one are shown in Figure 2.10; other time scales lead to results that are just intermediate to the three presented here.



(a) *#vod trace (1 month)* (b) *#yt trace (1 week)*

(c) *#vod trace (1 day)* (d) *#yt trace (4 hours)*

(e) *#vod trace (3 hours)* (f) *#yt trace (30 minutes)*

Figure 2.10: *Comparison between the hit probability of the original trace and the global randomization at different time scales.*

Near the first time scale (one week for #yt and one month for #vod) and beyond, all time scales have a request correlation structure that approaches the one observed in the full trace, and thus its hit probability differs significantly from that of the global randomization. Indeed, already at this time scale,

the MAPEs are of 5.3% and 11.6% in the `#yt` and `#vod` datasets, respectively; around the second time scale (four hours for `#yt` and one day for `#vod`), we observe a decrease in the discrepancies as the MAPEs are 5.0% and 2.3% in the `#yt` and `#vod` case, respectively. Though we see that the correlation structure does not influence strongly the hit probability, we remark again that the underestimation happens in the left side of the curves which corresponds to practical cache sizes. Finally, for traces around the last time scale (half hour for `#yt` and three hours for `#vod`), the MAPE are 1.4% and 2.3% for `#yt` and `#vod`, respectively, and we thus conclude that there are no significant structures between requests at this time scale.

Gained Insights

The results of the semi-experiments lead us to three main conclusions:

I1: At large time scales, the correlation structure of the whole request process is not negligible, in terms of hit probability. Additionally, we infer that most of the correlation comes from the fact that all requests for the same document are grouped within its lifespan.

I2: The document publications exhibit a correlation structure that does not have a significant impact on the hit probability. In particular, we deduce that document arrivals to the catalog can be modeled by an homogeneous Poisson process without losing accuracy on the estimation of the hit probability.

I3: For a given document, the request process within its lifespan exhibits some structure, but with little impact of the hit probability. Thus, for a given document, we can approximate the requests sequence by an homogeneous Poisson process defined on the lifespan of the document while still preserving the hit probability.

2.3 Definition of the Traffic Model

We build our model for the document request process by following a top-down approach (see Figure 2.11):

- on the top level, we consider a ground process Γ^g, hereafter called **catalog arrival process**; this point process dictates the consecutive arrivals of documents to the catalog. In our model, Γ^g is assumed to be a homogeneous Poisson process with constant intensity γ, according to insight I2.
- let a be the catalog arrival time of a document dictated by the process Γ^g. This event generates a **document request process** ξ_a determined by two random variables: the popularity R_a and the lifespan L_a. Specifically, following insight I3, we assume the process ξ_a to be Poisson with

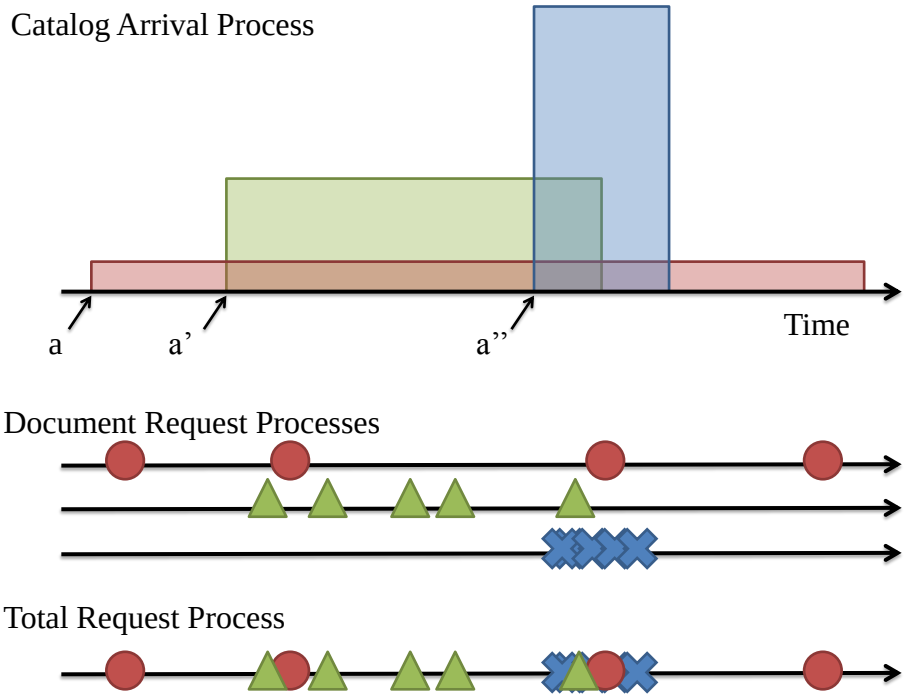


Figure 2.11: *Sample of the document arrival and request.* ***Top:*** *Boxes represent the lifespan and popularity by their width and height.* ***Bottom:*** *Sample of document request processes. Their superposition generates the total request process.*

intensity function R_a on interval $[a, a + L_a]$ and zero otherwise. We assume that the sequence $(R_a, L_a)_{a\in\Gamma^g}$ is almost surely i.i.d. and that the average number of requests for a document

$$\Lambda_a = R_a \cdot L_a$$

is almost surely finite;

- finally, the superposition of all processes ξ_a for all $a \in \Gamma^g$ generates the **total request process**

$$\Gamma = \sum_{a\in\Gamma^g} \xi_a$$

that contains the requests to all documents.

The point process Γ is a marked Poisson-Poisson cluster process [15, Sec. 6.3] because both the ground process Γ^g and the individual request processes $(\xi_a)_{a\in\Gamma^g}$ are Poisson. The marks indicate the specific document being requested at that time. Additionally, the process Γ can be regarded as a marked

Cox process [15, Sec. 6.2], where the random intensity function is given by the shot noise process

$$S(t) = \sum_{a \in \Gamma^g} R_a \cdot \mathbb{1}\{a \leq t \leq a + L_a\}$$

for $t \in \mathbb{R}$. Note that, since the ground process Γ^g and the sequence $(R_a, L_a)_{a \in \Gamma^g}$ are stationary, then the same holds for $S(t)$.

As show in Figure 2.11, the intensity S is superposition of "box" functions that are the request rate functions for each document. Thus, from now on we refer to this process as the *Box model*.

2.4 Validation

We now assess the validity of the Box model for the calculation of the hit probability. For this objective, we first obtain estimations of the model parameters in each dataset. Then, we plug these estimates into the theoretical formulas for the hit probability (see Chapter 3) to obtain an estimation of the cache performance. Finally, we compare the values predicted by the model to those obtained by a direct simulation of a LRU cache fed with the traces.

Estimation of the model parameters

Let K the number of observed documents. For document k, where $1 \leq k \leq K$, we denote by N_k its number of requests and by $(\Theta_1^k, \ldots, \Theta_{N_k}^k)$ its request sequence observed in data. Our aim is to estimate:

- The catalog arrival intensity γ.
- The distribution of the popularity-lifespan pair (R, L).

Let W be the size of the observation window. Then, the intensity γ is readily estimated by

$$\widehat{\gamma} = \frac{K}{W}.$$

Concerning the distribution of the popularity-lifespan pair (R, L), we first suppose without loss of generality that the data $(\{\Theta_1^k, \ldots, \Theta_{N_k}^k\})_{k=1}^K$ is in decreasing order of number of requests N_k. Then, to estimate this distribution, we use the point measure

$$\frac{1}{K_2} \sum_{k=1}^{K_2} \delta_{\widehat{R}_k, \widehat{L}_k}, \tag{2.2}$$

where K_2 is the number of documents with at least two requests and $(\widehat{R}_k, \widehat{L}_k)_{k=1}^{K_2}$ is a sequence of popularity-lifespan estimators for each document. Although these estimators are available only for

documents with more than 2 requests, we will see later that we can incorporate the rest of the information into the hit probability estimation. We estimate the lifespan of any document with $N_k \geq 2$ by

$$\widehat{L}_k = (\Theta^k_{N_k} - \Theta^k_1) \times \frac{N_k + 1}{N_k - 1}.$$

This estimator is unbiased since, that under the model assumptions, we have

$$\mathbb{E}\Big[\Theta^k_{N_k} - \Theta^k_1\Big] = \frac{N_k - 1}{N_k + 1} \times L_k.$$

Regarding popularity, we could give the crude estimate $N/\widehat{L}$, but our sample is biased by the fact that we collect only documents with at least one request. To take this bias into account, recall that N_k is a Poisson random variable with mean $R_k L_k$, given $N_k \geq 1$. We thus estimate the request rate R_k by

$$\widehat{R}_k = N'_k/\widehat{L}_k$$

where N'_k verifies equation

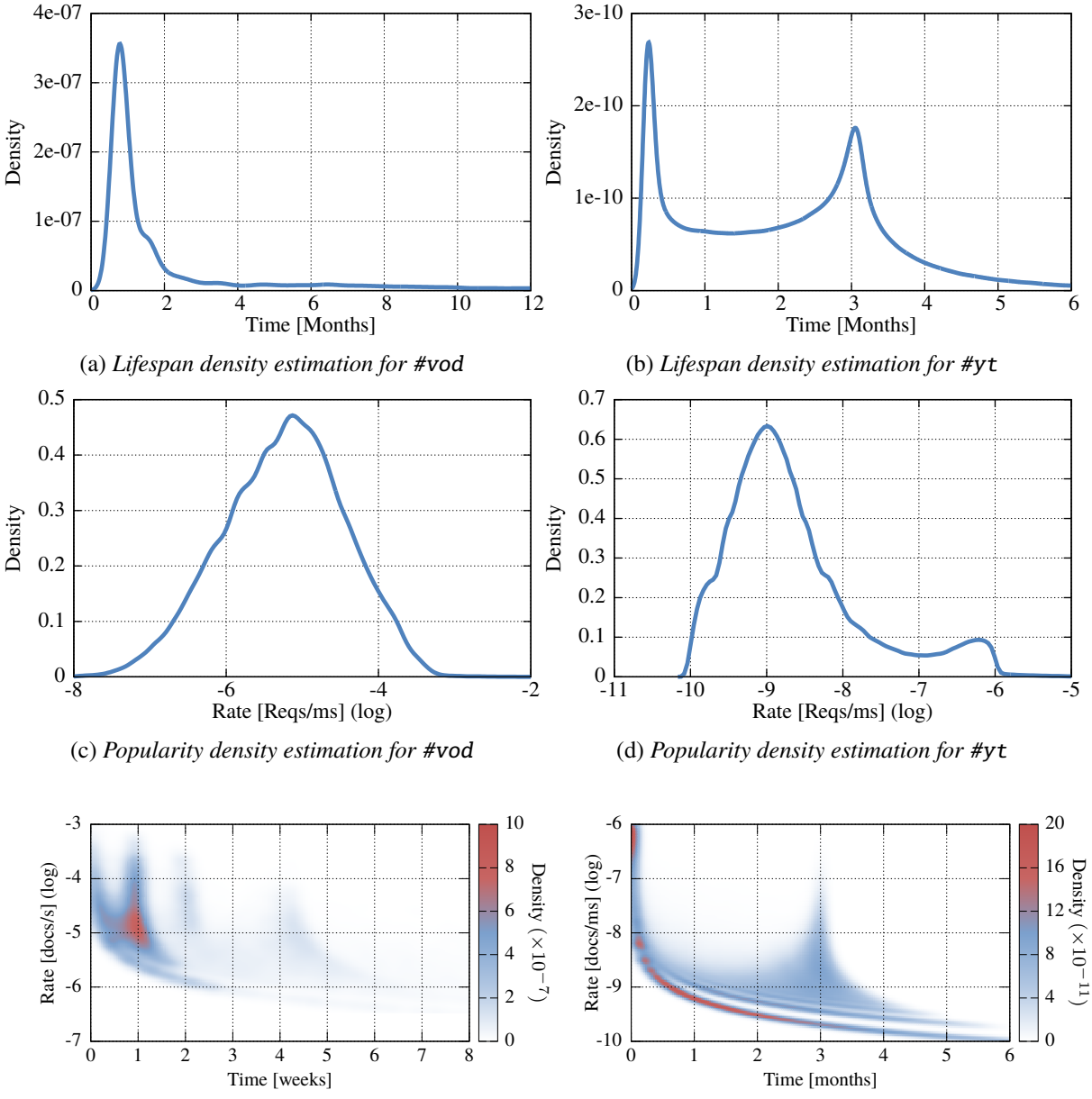
$$\frac{N'_k}{1 - e^{-N'_k}} = N_k.$$

The latter is shown to have a unique positive solution and we note that in practice that we can take $N'_k \approx N_k$ for N_k greater than 10.

Figures 2.12a and 2.12b show kernel density approximations for the lifespan distribution for each dataset. Note that the lifespan estimation formula yields a positive density for values larger than the observation window, especially for documents with a small number of requests. Also, in the `#yt` data, we observe a probability mass accumulation effect near the mark of three months, which is precisely the size of the observation window. This is a truncation effect and it is a sign that the lifespan of a video may be far longer than our current observation window in this dataset. As regards the `#vod` data, most documents have a lifespan shorter than one month. This corresponds to the numerous catch-up TV programs. The remaining documents have a different distribution, with lifespans varying on the range of a few weeks to the observation period (3.5 years). Due to the large observation period, the truncation effect is not visible.

As for the popularity distribution (Figures 2.12c and 2.12d), we see that the mass is distributed over many orders of magnitude which suggests a heavy tailed distribution. Again in the case of the `#yt` trace, we observe the censoring effect appearing at the left end of the distribution.

Finally the estimation for the joint distribution of the pair $(\log R, L)$ is shown in Figures 2.12e and 2.12f, with a focus on small values of the lifespan for the `#vod` data. In both cases, we conclude from the empirical densities that L and R are not independent random variables. Finally, the presence



(a) *Lifespan density estimation for* `#vod`

(b) *Lifespan density estimation for* `#yt`

(c) *Popularity density estimation for* `#vod`

(d) *Popularity density estimation for* `#yt`

(e) *Joint density estimation for* `#vod`

(f) *Joint density estimation for* `#yt`

Figure 2.12: *Popularity-Lifespan kernel density estimations*

of managed catch-up TV documents in the `#vod` data is visible; the marginal shows density peaks at values of 1, 2 and 4 weeks, corresponding to the duration for which broadcasts remain available.

Hit Probability Estimation

In order to calculate the hit probability, we use the estimation of the model parameters with the theoretical formulas that will be deduced in Chapter 3.

We readily express the hit probability in terms of the number of misses of a document μ_C:

$$q_C = 1 - \frac{\mathbb{E}[\mu_C]}{\mathbb{E}[N]} = 1 - \frac{\mathbb{E}[\mu_C]}{\mathbb{E}[N]}.$$

We will show in Chapter 3 that the average number of misses $\mathbb{E}[\mu_C]$ can be approximatively written as

$$\mathbb{E}[\mu_C] \approx m(t_C)$$

where

$$m(t) = \mathbb{E}\left[\left(1 - e^{-RL}\right) \mathbb{1}_{L \leq t} + \left(1 - e^{-Rt} + R(L-t)e^{-Rt}\right) \mathbb{1}_{L > t}\right] \tag{2.3}$$

and t_C is the characteristic time from the Che approximation. Specifically, in the present case, we have

$$t_C = \Xi^{-1}(C)$$

where Ξ is the average number of different documents requested in $[0, t]$. We will see in Chapter 3 that Ξ is simply related to m. In fact, we have in general that

$$\Xi(t) = \gamma \int_0^t m(u)\, \mathrm{d}u,$$

which, for the Box model, gives

$$\begin{aligned} \Xi(t) = \gamma\, \mathbb{E}\left[\left(2t + \left(1 - e^{-Rt}\right)\left(L - t - \frac{2}{R}\right)\right) \mathbb{1}_{L \geq t}\right] \\ + \gamma\, \mathbb{E}\left[\left(2L + \left(1 - e^{-RL}\right)\left(t - L - \frac{2}{R}\right)\right) \mathbb{1}_{L < t}\right]. \end{aligned} \tag{2.4}$$

As we have noted in the previous section, the estimators for the lifespan and popularity are not available for documents with only one request. However, this sub-sample can have a considerable size as evidenced in the `#yt` trace where it amounts to 58%. Thus, we cannot neglect this subset in a direct application of the hit probability formulas.

To incorporate this data, we use the approximation discussed in [30] where the set of documents requested only once is represented by a "noise" process. Let Ξ_1 (resp. Ξ_2) denote the mean function of that noise process (resp. the mean function associated with the "non-noise" part of the process), with $\Xi = \Xi_1 + \Xi_2$. We can separate the noise process from the rest of the request process and, using

a procedure similar to that for deducing Equation (2.4), we obtain an explicit formula for $\Xi_1(t)$ in the form

$$\Xi_1(t) = \gamma\,\mathbb{E}\left[\left(\frac{2}{R}\left(1-e^{-Rt}-Rte^{-Rt}\right)+(L-t)\left(Rte^{-Rt}\right)\right)\mathbb{1}_{L\geq t}\right] + \gamma\,\mathbb{E}\left[\left(\frac{2}{R}\left(1-e^{-RL}-RLe^{-RL}\right)+(t-L)\left(RLe^{-RL}\right)\right)\mathbb{1}_{L<t}\right].$$

Thus the "non-noise" part of the process Ξ_2 can be written as

$$\Xi_2(t) = \Xi(t) - \Xi_1(t) = \gamma\,\mathbb{E}[F(R,L,t)]\,, \tag{2.5}$$

where the function $F : \mathbb{R}_+^3 \mapsto \mathbb{R}_+$ is given by

$$F(R,L,t) = \left[2t(1-e^{-Rt})+\left(1-e^{-Rt}-Rte^{-Rt}\right)\left(L-t-\frac{4}{R}\right)\right]\mathbb{1}_{L\geq t} + \left[2L(1-e^{-RL})+\left(1-e^{-RL}-RLe^{-RL}\right)\left(t-L-\frac{4}{R}\right)\right]\mathbb{1}_{L<t} \quad .$$

Now, let K_1 be the number of documents with one request. We then estimate $\Xi_1(t)$ by the mean function of a homogeneous Poisson process:

$$\widehat{\Xi}_1(t) = K_1 \times \frac{t}{W}.$$

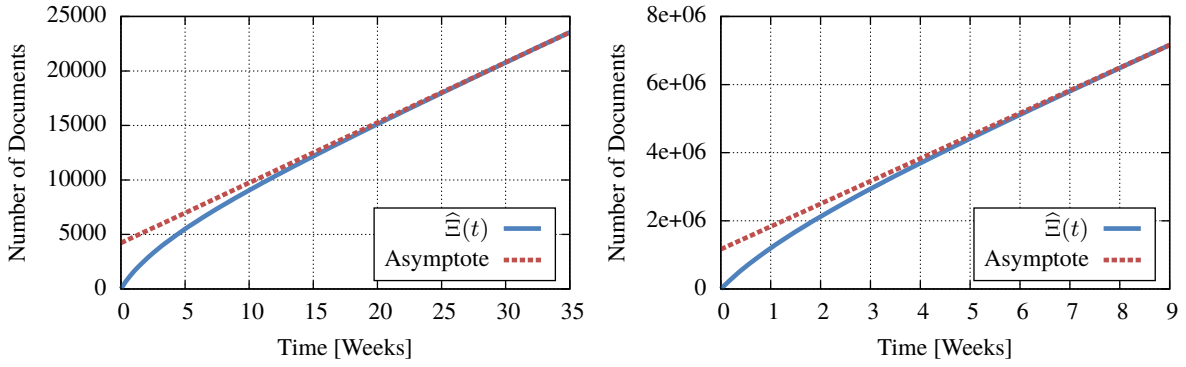
On the other hand, we estimate $\Xi_2(t)$ by the mean of the function F with respect to the point measure (2.2), that is,

$$\widehat{\Xi}_2(t) = \widehat{\gamma} \times \frac{1}{K_2}\sum_{i=1}^{K_2} F(\widehat{R}_i, \widehat{L}_i, t).$$

We then naturally set $\widehat{\Xi}(t) = \widehat{\Xi}_1(t) + \widehat{\Xi}_2(t)$ as the estimator of $\Xi(t)$ (see Fig. 2.13), and we use its inverse to estimate the characteristic time associated with the Che approximation, that is, $\widehat{t}_C = \widehat{\Xi}^{-1}(C)$. The latter inversion is carried on numerically.

With an estimation of the function Ξ in hand, we can proceed to estimate the hit probability q_C. In this case, we must similarly take the documents with just one request into account. First note that since the documents pertaining to the noise always produce misses, we can express the hit probability as

$$q_C = 1 - \frac{\mathbb{E}[\mu_C]}{\mathbb{E}[N]} = 1 - \frac{\mathbb{P}[N=1]+\mathbb{E}[\mu_C\cdot\mathbb{1}_{N\geq 2}]}{\mathbb{P}[N=1]+\mathbb{E}[N\cdot\mathbb{1}_{N\geq 2}]} = 1 - \frac{\frac{\mathbb{P}[N=1]}{\mathbb{P}[N\geq 2]}}{\frac{\mathbb{P}[N=1]}{\mathbb{P}[N\geq 2]}+\mathbb{E}[N\,|\,N\geq 2]} - \frac{\mathbb{E}[\mu_C\,|\,N\geq 2]}{\mathbb{E}[N\,|\,N\geq 2]+\frac{\mathbb{P}[N\geq 1]}{\mathbb{P}[N\geq 2]}}$$



(a) *#vod trace* (b) *#yt trace*

Figure 2.13: *Estimations for the average number of different documents mean function* $\widehat{\Xi}$*. The asymptote is shown to highlight the non-linear part at the beginning*

We now estimate each of the terms in the latter expression. Let G be the function such that

$$m(t) = \mathbb{E}[G(R, L, t)]$$

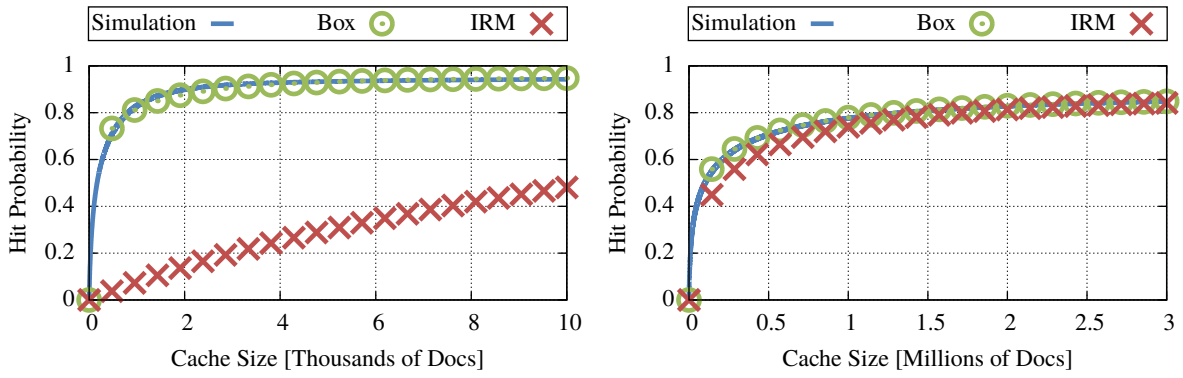
in Equation (2.3). We estimate $\mathbb{E}[\mu_C \mid N \geq 2]$ by taking the average of $G(\cdot, \cdot, \widehat{t_C})$ with respect to the point measure (2.2):

$$\mathbb{E}[\mu_C \mid N \geq 2] \approx \frac{1}{K_2} \sum_{i=1}^{K_2} G(\widehat{R}_i, \widehat{L}_i, \widehat{t_C}).$$

As to the term $\mathbb{E}[N \mid N \geq 2]$, it can be computed as the average number of requests in the corresponding sub-sample. Finally, the ratio $\mathbb{P}[N = 1] / \mathbb{P}[N \geq 2]$ is estimated by

$$\frac{\mathbb{P}[N = 1]}{\mathbb{P}[N \geq 2]} \approx \frac{K_1}{K_2}.$$

Using the above estimators, we can eventually compare the hit probability derived from the Box model to that obtained by simulation for each trace, as depicted in Figure 2.14. For comparison purpose, we provide also the estimation of the hit probability obtained by the Che approximation when the request process is assumed to be IRM. For the `#yt` traffic, the Box model improves the accuracy by one order of magnitude compared to the estimation with an IRM process, with respective MAPE of 0.5% and 4.1%. For the `#vod` traffic, the improvement is even more spectacular, due to the large duration of the trace. The IRM is far from estimating properly the hit probability with a MAPE of 17.2% (this value is significantly decreased by including the tail of the curve, not plotted here, and where the IRM converges towards the correct value). On the other hand, the Box model estimates accurately the hit probability,



(a) *#vod trace.* (b) *#yt trace.*

Figure 2.14: *Results of a simulation of the traces versus the fittings for the Che approximation with the Box model and IRM*

with a MAPE of 0.6%. These results effectively validate our proposed model.

2.5 Conclusion

The semi-experiments provided evidence that, at sufficiently large scales, the catalog dynamics has a non-negligible impact on the LRU cache performance. In consequence, simplistic models such as IRM make inaccurate prediction on the hit probability. The semi-experiments also shed light on the assumptions we can safely make for a model that exhibits catalog dynamicity. We consequently proposed and validated a simple yet accurate traffic model that characterize each document via an intrinsic popularity and a lifespan, and the catalog publications via a single rate.

In Chapter 3, we will justify the hit probability formulas we have used in the previous validation process. However, we will perform the analysis for a general class of cluster point processes that includes the Box model. We obtain these estimates by asymptotic methods and rigorously justify the Che approximation. The error of this approximation is also quantified.

The parameter fitting methods we have used here are ad-hoc and have the undesirable property of not being available for a potentially large portion of the dataset. This is due to the fact that the popularity and lifespan distributions are hidden parameters of the model, since we observe a random process that depends on random unobserved parameters. The standard way to treat this kind of problems is the Maximum Likelihood method. In Chapter 4, we apply this method to estimate the content popularity for a simpler model that can be seen as intermediate between the IRM and the Box model.

Chapter 3

Hit Probability Analysis

In this chapter, we develop a rigorous mathematical analysis for estimating the hit probability of an LRU cache fed by a cluster process. Specifically, we analyze a class of cluster processes such that:

- Each cluster is a Cox point process with almost surely a finite number of points.
- The ground process is a homogeneous Poisson process.

In the case when the clusters are mixed Poisson processes with a finite lifespan, we recover the Box model proposed in Chapter 2.

The arguments for our analysis are more clearly carried for the document miss probability p_C rather than for hit probability q_C. From now on, we thus analyze p_C, the hit probability being given $q_C = 1 - p_C$. Also note that

$$p_C = \frac{\mathbb{E}[\mu_C]}{\mathbb{E}[N]},$$

where μ_C and N the number of misses and requests for a document respectively. In consequence, the problem reduces to estimate the expectation $\mathbb{E}[\mu_C]$. We perform this task in three steps:

- The first step is to set up a probability space in which we tag a document to be examined independently from the rest of the traffic. This is advantageous since the eviction of a document from the cache depends only on events occurring in the rest of the traffic.
- In this setup we proceed to the second step, which is to deduce an integral formula for the quantity $\mathbb{E}[\mu_C]$. Due to all stationarity and independence structures of the model we obtain an intuitive formula, namely

$$\mathbb{E}[\mu_C] = \mathbb{E}[m(T_C)]\,.$$

This equality decouples the contributions of the tagged document and the rest of the process: First, given $t > 0$, the quantity $m(t)$ is the average number of misses of the tagged document in a t-TTL cache. Secondly, the *exit time* T_C is a random variable that measures the elapsed time between a document request and its eviction and for $t > 0$; deconditioning with respect to $T_C = t$ gives the formula $\mathbb{E}[\mu_C]$. Additionally, we prove that the exit time T_C is a first passage time for an inhomogeneous Poisson process with known mean function.

- The final step is to approximate the latter expectation. For this, we first consider a relevant scaling of model parameters. Then, under this scaling, we rigorously derive an asymptotic expansion of the expectation. This asymptotic expansion proves that the miss probability can be evaluated by approximating the exit time by the characteristic time of the "Che approximation". Moreover, we quantify the error of the approximation and show that it is of order $1/C$ for large C.

Additionally, we further validate these results empirically by comparing them to the hit probability obtained via the simulation of a cache fed by the Box model.

We start this chapter by first specifying the model in detail and its notation, and then proceed to our three step analysis.

3.1 Cluster Process Model

Our request model consists in a cluster point process on the real line $\mathbb{R}$ (see Figure 3.1). The associated ground process Γ^g, hereafter called **catalog arrival process**, dictates the consecutive arrivals of documents to the catalog. In the present setting, we consider Γ^g to be an homogeneous Poisson process with rate γ; we will denote any of its arrival times by the variable a.

The cluster at time a, denoted by ξ_a, represents the **document request process** for a document arriving to the catalog at that time. We consider ξ_a to be a Cox process directed by a non-negative stochastic intensity function λ_a. The intensity λ_a has the following properties:

- given the catalog arrival process Γ^g, the intensities $\lambda_a, a \in \Gamma^g$, are jointly independent;
- we consider *causal* random intensities λ_a, that is, each function $t \mapsto \lambda_a(t)$ is zero for $t < a$. This ensures that the requests in process ξ_a exist only after time a;
- the distribution of λ_a is "stationary" in the sense that for every arrival time $a \in \mathbb{R}$, the processes $\lambda_a(\cdot)$ and $\lambda_0(\cdot - a)$ have the same distribution.

These three conditions make the sampling of the sequence $(\lambda_a)_{a\in\Gamma^g}$ equivalent to an independent sample from a **canonical intensity function** λ with support in $[0, \infty)$, which is then shifted to each

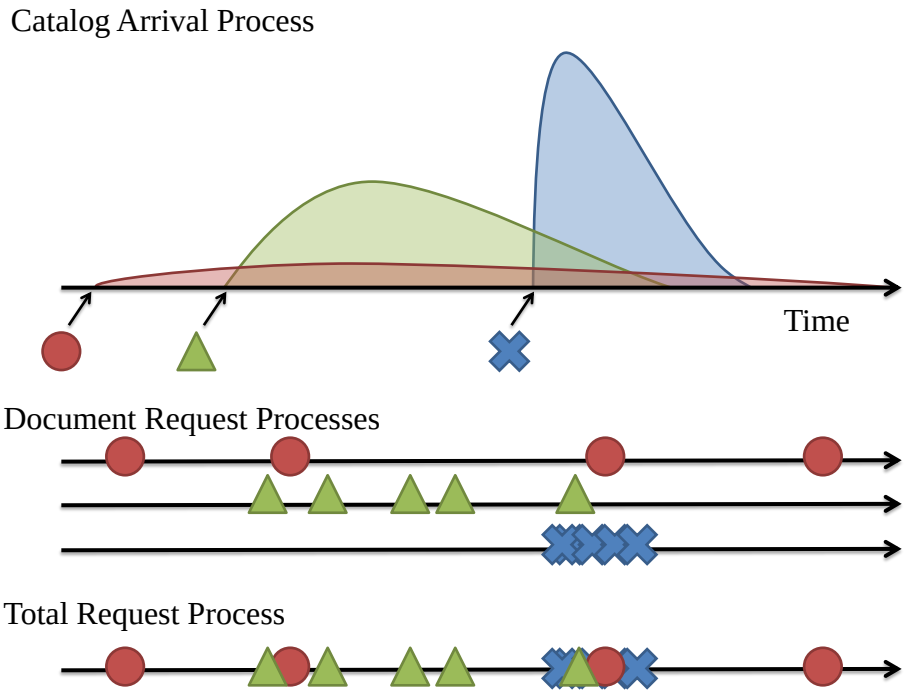


Figure 3.1: *A sample of the document arrival and request processes.* ***Top:*** *Each catalog arrival triggers a function representing the request intensity for the corresponding document.* ***Bottom:*** *A sample of the document request processes. Their superposition generates the total request process.*

arrival time a. We denote by Λ_a the associated **mean function** of λ_a, defined by

$$\Lambda_a(t) = \int_a^t \lambda_a(u)\mathrm{d}u,\ t > a.$$

For conciseness, we abuse the previous notation by denoting the **average number of requests** for a document arriving at time a as $\Lambda_a = \Lambda_a(\infty) \geq 0$ which we assume to be *finite* almost surely. We also denote by $\bar{\Lambda}_a$ the **complementary mean function**, that is $\bar{\Lambda}_a(t) = \Lambda_a - \Lambda_a(t),\ t > a$. When referring to the canonical document, which arrives at time zero, we simply remove the time index a. For example, $\bar{\Lambda}(t)$ represents the complementary mean function for the canonical document.

The superposition of all processes $\xi_a, a \in \Gamma^g$, generates the **total request process**

$$\Gamma = \sum_{a \in \Gamma^g} \xi_a$$

that represents the requests to all documents. Throughout the rest of the paper, we further assume that

$$\int_{-\infty}^{t} \mathbb{E}\Big[1 - e^{-(\Lambda(u)-\Lambda(u-t))}\Big]\, \mathrm{d}u < \infty \tag{3.1}$$

for any $t \in \mathbb{R}^+$. This is a sufficient and necessary condition for the process Γ to be well defined, in the sense that any compact set contains a finite number of points almost surely (see Theorem 6.3.III in [15]).

3.2 The Point of View of a Document

The key of our analysis is to *tag* one document from the system and treat the remaining process as an external *environment*. To this aim, we follow p. 279 in [16] and introduce the space $\mathcal{M}^{\#}(\mathbb{R})$ of point processes on $\mathbb{R}$; let $Q_{a,\nu}$ denote the local Palm distribution at point $(a, \nu) \in \mathbb{R} \times \mathcal{M}^{\#}(\mathbb{R})$ for the marked point process

$$\widetilde{\Gamma} = \sum_{a \in \Gamma^g} \delta_{a,\xi_a},$$

that is, the ground process Γ^g marked with the document request processes. Define then the mark-averaged Palm distribution $\overline{Q}_u$ on $\mathcal{M}^{\#}(\mathbb{R})$ by

$$\overline{Q}_u(\cdot) = \mathbb{E}[Q_{u,\xi_u}(\cdot)]\,.$$

For this distribution $\overline{Q}_u$, the process has the structure given by the following proposition (see Figure 3.2 for illustration).

Proposition 1
Under the distribution $\overline{Q}_u$, the process $\tilde{\Gamma}$ has almost surely a point at time u. Furthermore:

- *the distribution of the mark ξ_u is kept the same;*
- *the distribution of the remaining process $\tilde{\Gamma} \setminus \delta_{u,\xi_u}$ is the same than that of the original process $\tilde{\Gamma}$;*
- *the mark ξ_u and the process $\tilde{\Gamma} \setminus \delta_{u,\xi_u}$ are independent.*

Proof. We provide a quick proof using from the Slivnyak-Mecke Theorem (see [16], Prop. 13.1.VII). The latter theorem characterize the Laplace functional of Poisson point processes under their Palm distributions. In our case, for (u, ν) in $\mathbb{R} \times \mathcal{M}^{\#}(\mathbb{R})$, the Laplace functional $\mathcal{L}_{u,\nu}$ of $\tilde{\Gamma}$ under the Palm distribution $Q_{u,\nu}$ can be expressed by

$$\mathcal{L}_{u,\nu}[f] = e^{-f(u,\nu)} \cdot \mathcal{L}[f]$$

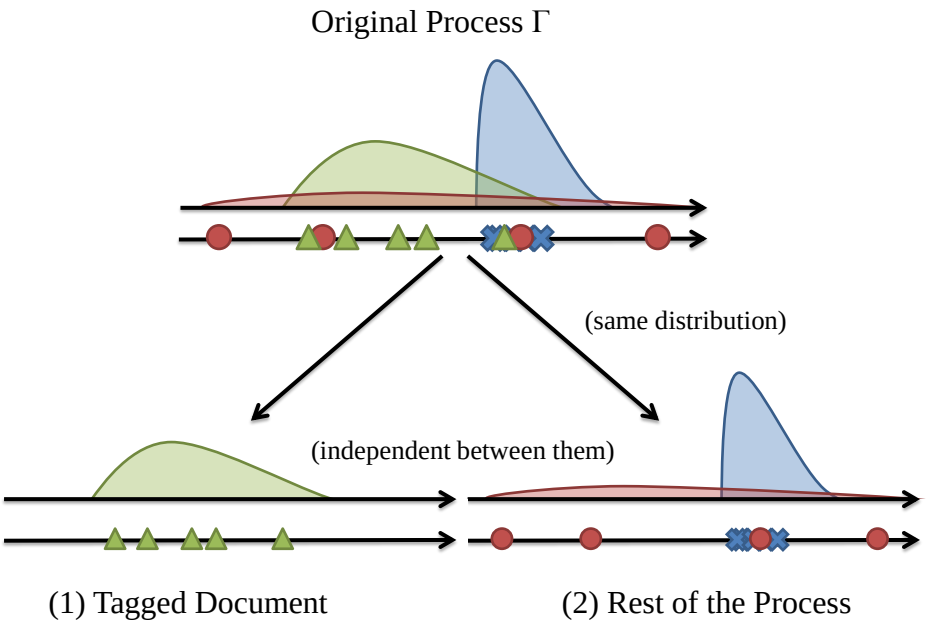


Figure 3.2: *Illustration of request process ξ under the averaged Palm distribution. The original process is decomposed into: (1) the tagged document and (2) the rest of the process. They are mutually independent and the rest of the process has the same distribution as the original.*

for any measurable function $f : \mathbb{R} \times \mathcal{M}^{\#}(\mathbb{R}) \to \mathbb{R}^{+}$, where $\mathcal{L}$ is the Laplace functional on the original probability space. The Laplace functional $\mathcal{L}_u$ under $\overline{Q}_u$ is consequently given by

$$\mathcal{L}_u[f] = \mathbb{E}[\mathcal{L}_{u,\xi_u}[f]] = \mathbb{E}\left[e^{-f(u,\xi_u)}\right] \mathcal{L}[f]\,.$$

Note that the expectation in the right-hand side is the Laplace functional of the point process δ_{u,ξ_u}. Since Laplace functionals characterize point processes, the conclusion follows. □

The properties claimed in Proposition 1 allow us to set a probability space where we have a document arrival almost surely at time $a = 0$. From now on, we consider this document as the **tagged** document, and the complementary process will be simply called **the rest**. In the next section, we will see that the considered LRU caching discipline, joint with the independence of the tagged document from the rest, allow us to derive a general integral formula for the miss probability.

3.3 A General Integral Formula

As stated in the previous section, we will consider a tagged document at time zero, so that its associated distribution is the canonical one. For a LRU cache with size C, let N and μ_C be the random number of requests and number of misses for the tagged document. The total miss probability is defined by

$$p_C = \frac{\mathbb{E}[\mu_C]}{\mathbb{E}[N]}\,,$$

which is also the average per-document hit ratio μ_C/N under the size biased distribution of N. Since N is a mixed Poisson variable with random mean Λ, we have

$$\mathbb{E}[N] = \mathbb{E}[\mathbb{E}[N \mid \Lambda]] = \mathbb{E}[\Lambda]\,,$$

and it is left to study μ_C.

Let $(\Theta_j)_{j=1}^N$ be the sequence of request times for the tagged document, with the understanding that it is the empty set if $N = 0$. The first request being always a miss, the number of misses can be written as

$$\mu_C = \mathbb{1}\{N \geq 1\} + \mathbb{1}\{N \geq 2\} \sum_{j=2}^{N} \mathbb{1}\{\text{Request at } \Theta_j \text{ is a miss}\}. \tag{3.2}$$

Under the LRU policy, a document requested at time s will be erased from the cache at the first time, after the last request for this document, that C distinct other documents have been requested.

For $s \in \mathbb{R}$, let $X^s = (X_t^s)_{t \geq s}$ denote the process that counts the number of distinct documents in the **rest** of the process requested on the interval $[s, t]$. We also define the family of **exit times** $(T_C^s)_{s \in \mathbb{R}}$ as the first passage time to level C of X^s. This quantity is the time that a document requested at time s can spend in the cache before being evicted. Denoting by $F^s(\xi_a)$ the first arrival time of ξ_a in $[s, \infty)$, the process X^s and exit times T_C^s can be expressed as

$$\begin{cases} X_t^s = \#\{(a, \xi_a) \text{ in } \tilde{\Gamma} \setminus \delta_{0,\xi_0} : F^s(\xi_a) \leq t\}\,, \quad t \geq s\,, \\ \\ T_C^s = \inf\{t \geq s : X_t^s = C\}\,. \end{cases} \tag{3.3}$$

The above definitions allow us to express the miss events as

$$\{\text{Request at } \Theta_j \text{ is a miss}\} = \{X_{\Theta_j}^{\Theta_{j-1}} \geq C\} = \{\Theta_j > T_C^{\Theta_{j-1}}\}\,, \quad j \geq 2\,,$$

since such a miss occurs if and only if at least C distinct other documents have been requested in the

interval $[\Theta_{j-1}, \Theta_j]$. Hence (3.2) can be written as

$$\mu_C = \mathbb{1}\{N \geq 1\} + \mathbb{1}\{N \geq 2\} \sum_{j=2}^{N} \mathbb{1}\left\{\Theta_j > T_C^{\Theta_{j-1}}\right\}. \tag{3.4}$$

To proceed further, we study the consequences of the structure of the cluster point process on the structure of the families X^s and T_C^s.

Proposition 2 (Characterization of X^s and T_C^s)
The process $X^s = (X_t^s)_{t \geq s}$ defined by (3.3) is an inhomogeneous Poisson process with mean function

$$\Xi^s(t) = \mathbb{E}[X_t^s] = \gamma \int_{-\infty}^{t} \mathbb{E}\left[1 - e^{-(\Lambda_a(t) - \Lambda_a(s))}\right] \mathrm{d}a\,, \quad t \geq s\,. \tag{3.5}$$

In particular, $T_C^s - s \stackrel{d}{=} T_C$, where $T_C = T_C^0$ is the exit time of a document requested at time zero.

Proof. By condition (3.1), we have $\Xi^s(t) < \infty$ for $t \geq s$. Now, the process $(X_u^s)_{s \leq u \leq t}$ is defined by counting the points (a, ξ_a) in the rest $\widetilde{\Gamma} \setminus \delta_{0,\xi_0}$ such that $F^s(\xi_a)$ falls in $[s, t]$; on the other hand, for $h \geq 0$, the increment $X_{t+h}^s - X_t^s$ counts only those points such that $F^s(\xi_a)$ falls in $(t, t+h]$. Since the corresponding two subsets of $\mathbb{R} \times \mathcal{M}^{\#}(\mathbb{R})$ are disjoint and $\widetilde{\Gamma} \setminus \delta_{0,\xi_0}$ is Poisson, we conclude that X^s has independent increments. In consequence, since X^s is a counting process, it is a inhomogeneous Poisson process.

The mean function for this process is then given by

$$\mathbb{E}[X_t^s] = \mathbb{E}\left[\sum_{a \in \Gamma^g} \mathbb{1}\{F^s(\xi_a) \in [s,t]\}\right] = \mathbb{E}\left[\sum_{a \in \Gamma^g} \mathbb{1}\{\xi_a[s,t] \geq 1\}\right].$$

Formula (3.5) follows from the latter expression and the fact that the mean measure η of $\widetilde{\Gamma} \setminus \delta_{0,\xi_0}$ is defined by

$$\eta([t_1, t_2] \times B) = \gamma \int_{t_1}^{t_2} \mathbb{P}[\xi_a \in B]\, \mathrm{d}a$$

where B is any Borelian of $\mathcal{M}^{\#}(\mathbb{R})$. □

Equation (3.4), Proposition 2, and the independence between the tagged document and the rest of the process now yield an integral formula for $\mathbb{E}[\mu_C]$.

Proposition 3
The expected number of misses is given by

$$\mathbb{E}[\mu_C] = \mathbb{E}[m(T_C)] \tag{3.6}$$

where $T_C = T_C^0$ denotes the exit time for a document requested at time zero (see (3.3)), and the function m is defined by

$$m(t) = \mathbb{E}\left[\int_0^\infty \lambda(u) e^{-(\Lambda(u+t)-\Lambda(u))}\,\mathrm{d}u\right], \quad t \geq 0\,. \tag{3.7}$$

Moreover, $\lim_{t\to\infty} \downarrow m(t) = m_0$*, where* $m_0 = \mathbb{E}\big[1 - e^{-\Lambda}\big]$.

The proof of Proposition 3 will follow from the following lemma, which holds for a class of functionals of the holding times of a Poisson process.

Lemma 4 (Functionals of holding times)
Let ξ be an inhomogeneous Poisson process on $[0,\infty)$ with deterministic *intensity function λ. Let the mean function Λ satisfy $\Lambda(\infty) < \infty$, so that ξ has a finite random number N of points $(\Theta_j)_{j=1}^N$. Then, for any $F : \mathbb{R}^+ \to \mathbb{R}$,*

$$\mathbb{E}\left[\mathbb{1}\{N \geq 2\} \sum_{j=2}^{N} F(\Theta_j - \Theta_{j-1})\right] = \int_0^\infty \mathrm{d}w\, F(w) \int_0^\infty \mathrm{d}u\, \lambda(u)\lambda(u+w) e^{-(\Lambda(u+w)-\Lambda(u))}\,.$$

We refer to Section 3.7 for the proof of this lemma.

Proof of Proposition 3. Since N is a mixed Poisson random variable with random mean Λ, the expectation of the first term on the r.h.s. of (3.4) is $\mathbb{P}[N \geq 1] = \mathbb{E}\big[1 - e^{-\Lambda}\big] = m_0$. For the second term on the r.h.s. of (3.4), since the family T_C^s for $s \geq 0$ is defined on the rest of the process and thus is independent from the request process $\xi = \sum_{j=1}^N \delta_{\Theta_j}$ for the tagged document, we have

$$\begin{aligned}
\mathbb{E}\left[\mathbb{1}\{N > 2\} \sum_{j=2}^{N} \mathbb{1}\left\{\Theta_j > T_C^{\Theta_{j-1}}\right\} \mid \xi\right] &= \mathbb{E}\left[\mathbb{1}\{n > 2\} \sum_{j=2}^{n} \mathbb{1}\left\{t_j > T_C^{t_{j-1}}\right\}\right]\Bigg|_{(n,t_1,\ldots,t_n)=(N,\Theta_1,\ldots,\Theta_N)} \\
&= \mathbb{E}\left[\mathbb{1}\{n > 2\} \sum_{j=2}^{n} \mathbb{1}\{t_j - t_{j-1} > T_C\}\right]\Bigg|_{(n,t_1,\ldots,t_n)=(N,\Theta_1,\ldots,\Theta_N)}
\end{aligned}$$

where the last equality follows from $T_C^s - s \overset{\mathrm{d}}{=} T_C$ (see Proposition 2). Taking the expectation and summing with the expectation of the first term yields

$$\mathbb{E}[\mu_C] = m_0 + \mathbb{E}\left[\mathbb{1}\{N > 2\} \sum_{j=2}^{N} \mathbb{1}\{\Theta_j - \Theta_{j-1} > T_C\}\right].$$

Since the canonical intensity λ and exit time T_C are independent from the request process of the tagged

document, Lemma 4 yields that

$$\begin{aligned}\mathbb{E}[\mu_C] &= m_0 + \mathbb{E}\left[\int_0^\infty \mathrm{d}w\, \mathbb{1}\{w > T_C\} \int_0^\infty \mathrm{d}u\, \lambda(u)\lambda(u+w) e^{-(\Lambda(u+w)-\Lambda(u))}\right] \\ &= m_0 + \mathbb{E}\left[\int_0^\infty \lambda(u) \left(e^{-(\Lambda(u+T_C)-\Lambda(u))}\right) \mathrm{d}u - \int_0^\infty \lambda(u) e^{-(\Lambda-\Lambda(u))}\, \mathrm{d}u\right] \\ &= \mathbb{E}\left[\int_0^\infty \lambda(u) \left(e^{-(\Lambda(u+T_C)-\Lambda(u))}\right) \mathrm{d}u\right],\end{aligned}$$

where we use for the last equality that, since $\Lambda(\infty) = \Lambda$ and $\Lambda(0) = 0$,

$$\int_0^\infty \lambda(u) e^{-(\Lambda-\Lambda(u))}\, \mathrm{d}u = \left[e^{-(\Lambda-\Lambda(u))}\right]_0^\infty = 1 - e^{-\Lambda}\,.$$

This last equation and dominated convergence imply that $\lim_{t\to\infty} \downarrow m(t) = \mathbb{E}\big[1 - e^{-\Lambda}\big]$, which concludes the proof. □

The above analysis would identically apply if the random variable T_C were deterministic and equal to some positive constant t. This would correspond to the cache discipline known as *Time to Live (TTL)*, where the cache evicts a document after a fixed amount of time t. Therefore, $m(t)$ is simply the average number of misses for a TTL cache of eviction time t. We can thus regard the number of misses in a LRU cache as a time randomization of the misses in a TTL cache.

Indeed, the integral formula (3.7) in Proposition 3 can by rewritten using integration by parts as

$$m(t) = \mathbb{E}\left[\int_0^\infty \lambda(u)\, e^{-(\Lambda(u)-\Lambda(u-t))}\, \mathrm{d}u\right]$$

which can be informally interpreted with as follows. The exponential term

$$e^{-(\Lambda(u)-\Lambda(u-t))}$$

is simply the conditional probability $\mathbb{P}[\xi[u-t,u] = 0 \,|\, \lambda]$. Thus a request at time u will contribute to the intensity of the miss process if there were no requests in the interval $[u-t,u]$, which is exactly a miss event in a t-TTL cache. This relationship between the miss probabilities of TTL and LRU caches has been already noted by Fofack et al. in [24].

3.4 An Asymptotic Expansion

It can be argued simply that $\mathbb{E}[\mu_C]$ has a finite limit as $C \to \infty$. In fact, first observe from formula (3.7) that as $t \to \infty$, $m(t)$ decreases monotonically to m_0 which is the average minimum number of misses;

secondly, from basic properties of Poisson processes, it can be asserted that the exit time T_C tends to infinity as $C \to \infty$. Applying formula (3.6) then provides the limit m_0 for $\mathbb{E}[\mu_C]$ when C grows large.

Another way to derive asymptotics for $\mathbb{E}[\mu_C]$ is to scale some system parameter with respect to C. Intuitively, the catalog arrival rate γ is a good candidate to be scaled by the cache size C, since one needs more storage in the cache to cope with an increasing document arrival rate. In the following, with help of the results of the previous sections, we will show that the problem is simply amenable to this setting and provide an asymptotic expansion for $\mathbb{E}[\mu_C]$ as C grows large.

To this aim, we first note from Proposition 2 that the canonical exit time T_C is the first passage time to level C of a inhomogeneous Poisson process with mean function $\Xi = \Xi^0$. To continue the analysis further, we first prove a key relation between m and Ξ.

Proposition 5

The functions m and Ξ satisfy the relation

$$\Xi'(t) = \gamma\, m(t)$$

for all $t \geq 0$.

The proof of this proposition is a consequence of various integration by parts and routine calculations. We thus defer it to Section 3.7.

By integration of the obtained identity with respect to time and by inversion, a consequence of the previous Proposition 5 is that

$$\Xi^{-1}(y) = M^{-1}\left(\frac{y}{\gamma}\right), \quad y \geq 0, \tag{3.8}$$

where

$$M(t) = \int_0^t m(s)\,\mathrm{d}s.$$

Besides, as observed in the beginning of the section, the exit time T_C is distributed as the first passage time to level C of an inhomogeneous Poisson process with mean function Ξ; it follows that T_C can be expressed by

$$T_C = \Xi^{-1}(\widehat{T}_C) \tag{3.9}$$

where $\widehat{T}_C$ is the first passage time to level C of an unitary homogeneous Poisson process, which has a Gamma$(C, 1)$ distribution. From Proposition 3 together with (3.9), we then derive that

$$\mathbb{E}[\mu_C] = \mathbb{E}[m(T_C)] = \mathbb{E}\Big[m(\Xi^{-1}(\widehat{T}_C))\Big]$$

and relation (3.8) eventually gives

$$\mathbb{E}[\mu_C] = \mathbb{E}\left[m\left(M^{-1}\left(\frac{\widehat{T}_C}{\gamma}\right)\right)\right]. \tag{3.10}$$

Now, recall that by the law of large numbers, we have $\widehat{T}_C/C \to 1$ as $C \to \infty$ almost surely. Thus, the latter formula suggests the scaling

$$C = \gamma\theta \tag{3.11}$$

in order obtain further asymptotics of $\mathbb{E}[\mu_C]$. This scaling relation is natural in the sense that by applying Little's law ([5], Section 3.1.2) to the cache system, we obtain $C = \gamma\,\mathbb{E}\big[T_C^{\text{in}}\big]$ where

$$T_C^{\text{in}} = \int_0^\infty \mathbb{1}\{\text{Object is in the cache at } t\}\,\mathrm{d}t$$

is the sojourn time of an object in the cache. Note that we do take into account the objects that do not have any requests as entering the system, but we set their sojourn time to $T_C^{\text{in}} = 0$. As a consequence, the asymptotic analysis under the scaling (3.11) amounts to fixing the average sojourn time $\theta = \mathbb{E}\big[T_C^{\text{in}}\big] = C/\gamma$ and the distribution of the canonical intensity function λ while letting C grow to infinity.

Applying scaling (3.11), relation (3.10) implies

$$\lim_{C\to\infty} \mathbb{E}[\mu_C] = m(M^{-1}(\theta)) = m(t_\theta); \tag{3.12}$$

in the following, the quantity t_θ will be called the **characteristic time** and the asymptotics of $\mathbb{E}[\mu_C]$ will be expressed in terms of it. In this aim, we first recall two basic results regarding the $\mathrm{Gamma}(C, 1)$ distribution.

Lemma 6

Define the random variable X_C by

$$X_C = \frac{\widehat{T}_C}{C}$$

where $\widehat{T}_C$ follows a $\mathrm{Gamma}(C, 1)$ distribution. Then

i) for any $C > 1$ and $\eta > 0$, we have

$$\mathbb{P}[|X_C - 1| \geq \eta] \leq 2e^{-C\cdot\varphi(1+\eta)}$$

where $\varphi(x) = x - 1 - \log x$ is the rate function of an exponential random variable of mean 1;

ii) for any $C > 1$ and $k > 1$, we have

$$\mathbb{E}\Big[(X_C - 1)^k\Big] = O(C^{-\lceil \frac{k}{2} \rceil}).$$

We refer to Section 3.7 for the proof. We now formulate our central result concerning the asymptotics for the average number of misses.

Theorem 7

Assume that the function m is twice continuously differentiable in $(0, \infty)$. Under the scaling $C = \gamma\theta$, we then have

$$\mathbb{E}[\mu_C] = m(t_\theta) + \frac{e(t_\theta)}{C} + o\left(\frac{1}{C}\right) \tag{3.13}$$

as $C \to \infty$, where the error term $e(t_\theta)$ is given by

$$e(t_\theta) = \left[\frac{\theta^2}{2m(t_\theta)^2}\left(m''(t_\theta) - \frac{m'(t_\theta)^2}{m(t_\theta)}\right)\right].$$

Proof sketch. From (3.10) and the scaling $C = \gamma\theta$ we obtain

$$\mathbb{E}[\mu_C] = \mathbb{E}\left[m\left(M^{-1}\left(\theta \times \frac{\widehat{T}_C}{C}\right)\right)\right].$$

Since a.s. $\widehat{T}_C/C \to 1$ as $C \to \infty$, the main idea of the proof is to apply a Taylor expansion around 1 to the function

$$f_\theta(\cdot) = m(M^{-1}(\theta \times \cdot)).$$

The bulk of the proof is devoted to the use of Lemma 6 and limit theorems to justify the limit and expectation exchange and obtain the error term e. See Section 3.7 for the details. □

The expansion in Theorem 7 justifies the accuracy of the estimations based on the Che approximation (see Appendix 6.1 for a derivation of the classic version of this heuristic). In fact, in the present setting, this heuristic consists in replacing the exit time T_C in (3.6) by the characteristic time $\widetilde{t}_C = \Xi^{-1}(C)$, therefore estimating $\mathbb{E}[\mu_C]$ by $m(\widetilde{t}_C)$. Now, under the scaling $C = \gamma\theta$, identity (3.8) entails that

$$\begin{aligned}\widetilde{t}_C = \Xi^{-1}(C) &= M^{-1}\left(\frac{C}{\gamma}\right) \\ &= M^{-1}(\theta) = t_\theta.\end{aligned}$$

Thus, the previous identity justifies this naming for t_θ as well. More importantly, the asymptotic expansion of $\mathbb{E}[\mu_C]$ in Theorem 7 shows that the error in the Che approximation is of order $1/C$,

for large C and fixed average sojourn time θ. We thus have explicitly quantified the accuracy of this approximation (see the conclusion for a more detailed explanation).

Remark 8

If the function m has higher order derivatives, the proof of Theorem 7 together with Lemma 6 allow us to derive higher order expansions of $\mathbb{E}[\mu_C]$ in powers of $1/C$. Specifically, to obtain an expansion at order n, we must expand f_θ to the $2n$-th order, since $\mathbb{E}\big[(X_C-1)^k\big]$ is $O(1/C^{\lceil \frac{k}{2} \rceil})$ by Lemma 6. We then eventually obtain

$$\mathbb{E}[\mu_C] = \sum_{k=0}^{2n} \frac{f_\theta^{(k)}(1)}{j!} \frac{\phi_k(C)}{C^k} + o\left(\frac{1}{C^n}\right)$$

where ϕ_k is a polynomial of degree $\lfloor k/2 \rfloor$, as shown in the proof of Lemma 6 (see Section 3.7).

Remark 9

Theorem 7 can be proved by purely analytical methods. Indeed, equation (3.19) can be written in integral form as

$$\mathbb{E}[\mu_C] = \frac{C^C}{\Gamma(C)} \int_0^\infty e^{-C(w-\log(w))} \frac{f_\theta(w)}{w}\, \mathrm{d}w.$$

after using variable change $w \mapsto w/C$. For large C, Theorem 7 then follows by expanding the above integral by means of the Laplace method (see (3.15) in [48]) and denominator $\Gamma(C)$ via the Stirling formula. This method yields the same expansion for $\mathbb{E}[\mu_C]$ but it is more complicated in that it involves the expansion of both numerator and denominator in powers of $\sqrt{C}$.

To conclude this section, we show that the smoothness assumptions for function m in Theorem 7 hold for a class of random intensities λ suitable for modeling purposes. This class includes the Box model and families used in related works [59] and is built by randomly scaling a deterministic shape function in both domain and range.

Proposition 10

Let $f \in \mathcal{C}^1(0,\infty)$ be a strictly positive unimodal function with $\int f = 1$, $\int f^2 < \infty$ and $\int |f'| < \infty$. Consider a pair (R, L) of positive random variables with smooth joint density such that $\mathbb{E}[R] < \infty$ and $\mathbb{E}[RL] < \infty$. If the canonical document request intensity is distributed as

$$\lambda(u) = R \cdot f\left(\frac{u}{L}\right), \quad u \geq 0, \tag{3.14}$$

then function m is $\mathcal{C}^2(0,\infty)$ with derivatives given by

$$\begin{cases} m'(t) &= -\mathbb{E}\left[R^2 L \int_0^\infty f(u) f\left(u+\frac{t}{L}\right) e^{-RL\left(F\left(u+\frac{t}{L}\right)-F(u)\right)}\,\mathrm{d}u\right], \\ m''(t) &= \mathbb{E}\left[R^3 L \int_0^\infty f(u) f\left(u+\frac{t}{L}\right)^2 e^{-RL\left(F\left(u+\frac{t}{L}\right)-F(u)\right)}\,\mathrm{d}u\right] \\ &\quad -\mathbb{E}\left[R^2 \int_0^\infty f(u) f'\left(u+\frac{t}{L}\right) e^{-RL\left(F\left(u+\frac{t}{L}\right)-F(u)\right)}\,\mathrm{d}u\right] \end{cases} \tag{3.15}$$

for $t>0$, where $F(u)=\int_0^u f(v)dv$.

We defer the proof of the latter proposition to Section 3.7.

Note that Proposition 10 only imposes mild conditions on the distribution of (R,L). The admitted shape functions f include exponential and power law decreasing profiles, and Gaussian curves restricted to $[0,\infty)$. In addition, the assumption of f being strictly positive on $[0,\infty)$ can be weakened to that of being positive only in a compact interval; this in turn implies that f' is not differentiable everywhere and the second derivative of m will thus contain additional terms from the integral of f'. These terms can be obtained by an integration by parts (see [25], Th. 3.36 for a generalized form).

One example of such a family with compact support is given by the box model, which can be constructed by simply taking $f=\mathbb{1}_{[0,1]}$. In this case, m and its derivatives reduce to

$$\begin{cases} m(t) &= \mathbb{E}\left[\left(1-e^{-RL}\right)\mathbb{1}_{L\le t} + \left(1-e^{-Rt}+R(L-t)e^{-Rt}\right)\mathbb{1}_{L>t}\right], \\ m'(t) &= -\mathbb{E}\left[R^2(L-t)e^{-Rt}\,\mathbb{1}_{L>t}\right], \\ m''(t) &= \mathbb{E}\left[(R^2+R^3(L-t))e^{-Rt}\,\mathbb{1}_{L>t}\right]. \end{cases} \tag{3.16}$$

We will use this model for a numerical illustration in the next section.

3.5 Validation

We here provide some numerical results to validate the accuracy of asymptotic expansion (3.13), by comparing it to the values obtained from the system simulation. In our experiments, we use the Box Model in which the canonical intensity function given by

$$\lambda(u) = R\cdot\mathbb{1}\{0\le u\le L\}, \quad u\ge 0,$$

where the random pair (R, L) represents the request rate and lifespan of a document. In view of (3.13), we obtain the zero order and first order approximations for the hit probability q_C, namely

$$q_C = 1 - p_C = 1 - \frac{\mathbb{E}[\mu_C]}{\mathbb{E}[\Lambda]} \approx \begin{cases} 1 - \dfrac{m(t_\theta)}{\mathbb{E}[\Lambda]}, & \text{0th Order} \\ 1 - \dfrac{m(t_\theta) + e(t_\theta)/C}{\mathbb{E}[\Lambda]}, & \text{1st Order} \end{cases} \tag{3.17}$$

where $\mathbb{E}[\Lambda] = \mathbb{E}[RL]$. In general, for a given distribution of (R, L), we cannot deduce from (3.16) explicit expressions for m, m', m'', M and M^{-1} . In particular, there are usually no formulas for t_θ in terms of θ. In consequence, we resort to numerical integration and inversion to obtain the hit probability estimates in (3.17).

As argued in Section 2.4, the distributions of variable R and L have a support ranging over many scales of magnitude, suggesting a heavy tailed nature. For our experiments, we consequently chose R and L to be distributed as independent Pareto-Lomax variables, with probability density $\alpha\sigma^\alpha/(\sigma + x)^{\alpha+1}$, $x > 0$, with respective parameters $(\alpha = 1.9, \sigma = 22.5)$ and $(\alpha = 1.7, \sigma = 0.07)$. Such values have been taken so that the simulation time is not excessive; they provide a "box" of average width 0.1 and height 25 with high volatility since neither R nor L have a finite variance.

We generated the request process associated with these intensity functions for various values of γ ranging from 10 to 1 000. For each request sequence, we simulated an LRU cache and obtained the empirical hit probability for various capacities C. To obtain reliable results, the heavy tailed nature of the input distributions makes the use stable-law central limit theorem necessary ([61], Th. 4.5.1). Specifically, there exists a so-called stable law $S_\alpha(\sigma, \beta, \mu)$ with scaling parameter σ and a constant K_α, such that

$$\lim_{n\to\infty} \frac{1}{K_\alpha} \frac{1}{n^{1/\alpha}} \sum_{i=1}^{n} (L_i - n\mathbb{E}[L]) = S_\alpha(1, 1, 0)$$

in distribution. The latter allows us to heuristically quantify the convergence rate for the law of large numbers by

$$\frac{1}{n} \sum_{i=1}^{n} (L_i - n\mathbb{E}[L]) \approx S_\alpha \left(\frac{K_\alpha}{n^{1-1/\alpha}}, 1, 0 \right)$$

for large n (in the present case, $\alpha = 1.7$ for variable L). We then chose the simulation time S such that the average number of observed documents $n = \gamma S \times \mathbb{E}\left[1 - e^{-RL}\right]$ is such that scaling parameter $K_\alpha/n^{1-1/\alpha}$ is smaller than 10^{-3} (such a value of n ensures the same accuracy for the request rate R with larger tail index $\alpha = 1.9$). Besides, we also chose S large enough to ensure that there is enough time for all observable documents to appear in the simulated trace.

We show in Figure 3.3 some of the resulting hit probability curves from these experiments, and observe that the zero order approximation in (3.17) is exact for $\gamma = 500$ already. The error incurred

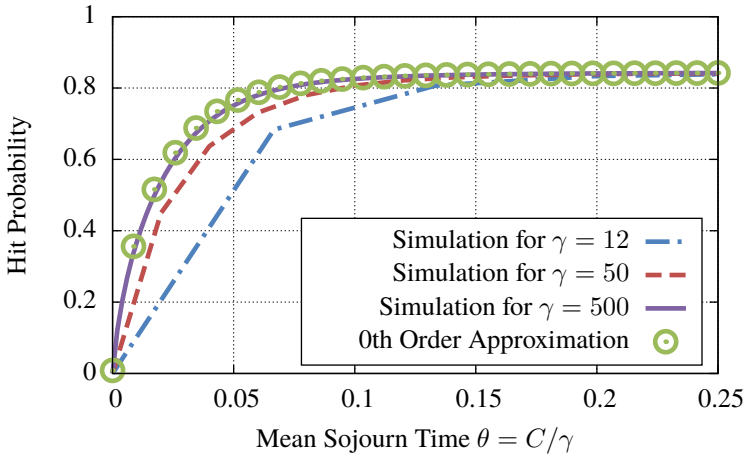
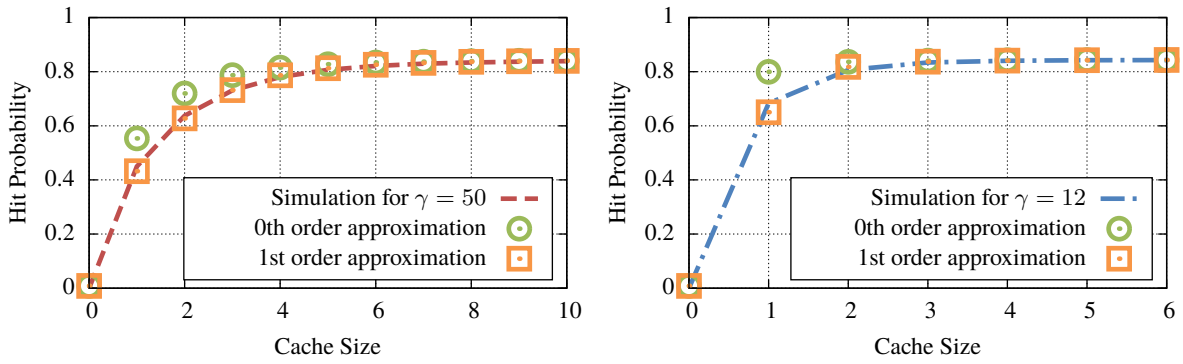


Figure 3.3: *Convergence to the 0th order approximation when $C \to \infty$ and $C = \gamma\theta$.*

by the approximation for lower γ can be corrected by using the first order approximation in (3.17), as shown in Figure 3.4a for $\gamma = 50$ (for even lower intensities, this correction might not be enough to approximate the real hit probability, as illustrated in Figure 3.4b for $\gamma = 12$; the higher order expansion of Remark 8 would then be needed).



(a) *Comparison with $\gamma = 50$* (b) *Comparison with $\gamma = 12$*

Figure 3.4: *Comparison between 0th and 1st approximations and the results of simulations.*

The above numerical results therefore illustrate the accuracy of the asymptotic expansion for the hit probability.

3.6 Conclusion

In this chapter, we have estimated the hit probability of a LRU cache for a traffic model based on a Poisson cluster point process. In this endeavor, we have built using Palm theory a probability space where a tagged document can be analyzed independently from the rest of the process. In the case of the LRU replacement policy, this property is key for the analysis, since it allowed us to derive an integral expression for the expected number of misses of the tagged object. Using this expression, we were able to obtain an asymptotic expansion of this integral for large C under the scaling $C = \gamma\theta$ for fixed $\theta > 0$.

Our framework and asymptotic analysis justify rigorously every step of the Che approximation (Section 1.3) for our traffic model. Indeed:

- Step **Che.1** is to assume that all exit times have the same distribution. In our case, this is justified the integral formula derived in Proposition 3, which tells us that the average number of misses can be expressed in terms of function m and the canonical exit time T_C. We justify the latter proposition using the Palm distribution of the system (Section 3.2) and applying Lemma 4.

- Step **Che.2** is to assume that the exit time is well approximated by the deterministic characteristic time. Strictly speaking, this assumption is false, since the exit times are random variables with diverging variance when $C \to \infty$.

  However, we have shown that in Theorem 7 that under the scaling $C = \gamma\theta$ the limiting hit probability depends on a characteristic time akin to the one in the classic Che approximation. Indeed, under the previous scaling, the characteristic time is defined by the solution to the following equation:

  $$C = \Xi(t) = \gamma\,\mathbb{E}\left[\int_{-\infty}^{t} 1 - e^{-(\Lambda_a(t)-\Lambda_a(0))}\,\mathrm{d}a\right].$$

  which is analogous to (1.2) in step **Che.2**.

- Finally, we have also shown in Theorem 7 that asymptotic hit probability satisfies

  $$q_C = 1 - \frac{m(t_\theta)}{\mathbb{E}[\Lambda]} + O\left(\frac{1}{C}\right) = \frac{\mathbb{E}\left[\int_0^\infty \lambda(u)\left(1 - e^{-(\Lambda(u+t_\theta)-\Lambda(u))}\right)\,\mathrm{d}u\right]}{\mathbb{E}\left[\int_0^\infty \lambda(u)\,\mathrm{d}u\right]} + O\left(\frac{1}{C}\right)$$

  which is analogous to Equation (1.3) in step **Che.3** but with the addition of an explicit quantification of the error incurred.

In addition to the latter justification, we have shown that the latter expansion is valid for a class of processes suitable for modeling purposes and that it is possible to estimate it numerically.

3.7 Technical Proofs

Proof of Lemma 4

Recall that, given that process ξ has k points, then the request times $(\Theta_j)_{j=1}^k$ have the distribution of the order statistics of a random variable with density $g(t) = \lambda(t)/\Lambda\ ,\ t \geq 0$ and distribution function G is given by $G(t) = \Lambda(t)/\Lambda\ ,\ t \geq 0$. Let $\overline{G}(t) = 1 - G(t)$ be the complement of G. From the standard order statistics theory, it is known that the distribution of the holding times $\Theta_j - \Theta_{j-1}, j > 1$, has the density $\tilde{g}_{k,j}$ given by

$$\tilde{g}_{k,j}(w) = \frac{k!}{(j-2)!(k-j)!} \int_0^\infty G^{j-2}(u)g(u)g(u+w)\overline{G}^{k-j}(u+w)\mathrm{d}u$$

for all $w \geq 0$. We can consequently write

$$\mathbb{E}[F(\Theta_j - \Theta_{j-1})\ \mid N = k] = \int_0^\infty \tilde{g}_{k,j}(w)F(w)\,\mathrm{d}w$$

which, in turn, allows us to write the desired expectation as

$$\begin{aligned}\mathbb{E}\left[\mathbb{1}\{N \geq 2\}\sum_{j=2}^{N} F(\Theta_j - \Theta_{j-1})\right] &= \sum_{k=2}^{\infty}\sum_{j=2}^{k} \mathbb{E}[F(\Theta_j - \Theta_{j-1}) \mid N = k]\, e^{-\Lambda}\frac{\Lambda^k}{k!} \\ &= \int_0^\infty F(w)\,\mathrm{d}w \sum_{k=2}^{\infty}\sum_{j=2}^{k} \tilde{g}_{k,j}(w)\, e^{-\Lambda}\frac{\Lambda^k}{k!}. \end{aligned} \tag{3.18}$$

Now, by the Binomial Theorem, we can write

$$\sum_{j=2}^{k} \frac{k!}{(j-2)!(k-j)!} G^{j-2}(u)\overline{G}^{k-j}(u+w) = k(k-1)[G(u) + \overline{G}(u+w)]^{k-2}$$

and thus, passing all the sums inside the integral in the right-hand side of (3.18), we obtain

$$\sum_{k=2}^{\infty}\sum_{j=2}^{k} \tilde{g}_{k,w}(w)e^{-\Lambda}\frac{\Lambda^k}{k!} = \Lambda^2 \int_0^\infty e^{-\Lambda(1-G(u)-\overline{G}(u+w))}G(u)g(u+w)\mathrm{d}u.$$

Note that

$$\Lambda(1 - G(u) - \overline{G}(u+w)) = \Lambda(G(u+w) - G(u)) = \Lambda(u+w) - \Lambda(u)$$

and

$$G(u)G(u+w) = \lambda(u)\lambda(u+w)/\Lambda^2.$$

Equation (3.18) together with the latter intermediate results eventually provides

$$\mathbb{E}\left[\mathbb{1}\{N \geq 2\} \sum_{j=2}^{N} F(\Theta_j - \Theta_{j-1})\right] = \int_0^\infty F(w)\,\mathrm{d}w \int_0^\infty \lambda(u)\lambda(u+w) e^{-(\Lambda(u+w)-\Lambda(u))}\,\mathrm{d}u$$

as claimed.

Proof of Proposition 5

We decompose the integral for $\Xi(t)$ into the contributions before and after time zero, giving

$$\begin{aligned}
\Xi(t) &= \gamma \int_{-\infty}^{0} \mathbb{E}\Big[1 - e^{-(\Lambda_a(t)-\Lambda_a(0))}\Big]\,\mathrm{d}a + \gamma \int_0^t \mathbb{E}\Big[1 - e^{-(\Lambda_a(t)-\cancelto{0}{\Lambda_a(0)})}\Big]\,\mathrm{d}a \\
&= \gamma \int_{-\infty}^{0} \mathbb{E}\Big[1 - e^{-(\Lambda(t-a)-\Lambda(-a))}\Big]\,\mathrm{d}a + \gamma \int_0^t \mathbb{E}\Big[1 - e^{-\Lambda(t-a)}\Big]\,\mathrm{d}a \\
&= \gamma\,(I_1(t) + I_2(t)).
\end{aligned}$$

Making the variable change $a \mapsto -a$ in the first integral $I_1(t)$ yields

$$I_1(t) = \int_{-\infty}^{0} \mathbb{E}\Big[1 - e^{-(\Lambda(t-a)-\Lambda(-a))}\Big]\,\mathrm{d}a = \int_0^\infty \mathbb{E}\Big[1 - e^{-(\Lambda(t+a)-\Lambda(a))}\Big]\,\mathrm{d}a.$$

Successively differentiating with respect to t and integrating by parts further gives

$$\begin{aligned}
I_1'(t) &= \int_0^\infty \mathbb{E}\Big[\lambda(t+a) e^{-(\Lambda(t+a)-\Lambda(a))}\Big]\,\mathrm{d}a \\
&= \mathbb{E}\Big[e^{-\Lambda(t)} - 1\Big] + \mathbb{E}\left[\int_0^\infty \lambda(a) e^{-(\Lambda(t+a)-\Lambda(a))}\mathrm{d}a\right] = \mathbb{E}\Big[e^{-\Lambda(t)} - 1\Big] + m(t)
\end{aligned}$$

where the last equality comes from Proposition 3. Now, for the second integral $I_2(t)$, the variable change $a \mapsto t - a$ yields

$$I_2(t) = \int_0^t \mathbb{E}\Big[1 - e^{-\Lambda(t-a)}\Big]\,\mathrm{d}a = \int_0^t \mathbb{E}\Big[1 - e^{-\Lambda(a)}\Big]\,\mathrm{d}a$$

so that $I_2'(t) = \mathbb{E}\big[1 - e^{-\Lambda(t)}\big]$. As a consequence, we obtain $\Xi'(t) = \gamma(I_1'(t) + I_2'(t)) = \gamma\, m(t)$ as claimed.

Proof of Lemma 6

i) This is the classic optimized exponential Markov inequality which is used for the upper bound in Cramer's large deviations Theorem, see [17, Th. 2.2.3, Remark (c)].

ii) We expand the k-th order central moment of X_C in terms of the known moments of $\widehat{T}_C$, giving

$$\begin{aligned}\mathbb{E}\Big[(X_C-1)^k\Big] &= \sum_{i=0}^{k}\binom{k}{i}\frac{\mathbb{E}\Big[(\widehat{T}_C)^i\Big]}{C^i}(-1)^{k-i}\\ &= \frac{1}{C^k}\sum_{i=0}^{k}\binom{k}{i}(-C)^{k-i}\frac{\Gamma(C+i)}{\Gamma(C)} = \frac{1}{C^k}\phi_k(C),\end{aligned}$$

where ϕ_k is a polynomial of degree at most k. As shown in [47], the polynomial ϕ_k is actually of degree $\lfloor k/2\rfloor$, which allows us to conclude.

Proof of Theorem 7

For fixed $\theta>0$, define the function f_θ by

$$f_\theta(z) = m(M^{-1}(\theta z)) = m(t_{\theta z}).$$

With the scaling $C=\gamma\theta$, Equation (3.10) can be then written as

$$\mathbb{E}[\mu_C] = \mathbb{E}\left[f_\theta\left(\frac{\widehat{T}_C}{C}\right)\right]. \tag{3.19}$$

Let again $X_C=\widehat{T}_C/C$ as in Lemma 6 and fix $\eta>0$. Write the expectation (3.19) as $\mathbb{E}[\mu_C]=A_C+B_C$ where

$$A_C = \mathbb{E}\big[f_\theta(X_C)\mathbb{1}_{|X_C-1|\geq\eta}\big], \quad B_C = \mathbb{E}\big[f_\theta(X_C)\mathbb{1}_{|X_C-1|<\eta}\big].$$

• To analyze A_C, recall that function m is bounded by $\mathbb{E}[\Lambda]<\infty$, and so is f_θ. Then, by Lemma 6 (i), we have

$$A_C \leq \mathbb{E}[\Lambda]\,\mathbb{P}[|X_C-1|\geq\eta] \leq 2\mathbb{E}[\Lambda]\,e^{-C\cdot\varphi(1+\eta)}$$

which shows, in particular, that $A_C=o(1/C)$.

• To analyze B_C, first write a Taylor expansion of f_θ around 1 at order two in the form

$$\begin{aligned}f_\theta(X_C) &= f_\theta(1)+f'_\theta(1)\,(X_C-1)+\frac{f''_\theta(Y_C)}{2}\,(X_C-1)^2\\ &= h_\theta(X_C)+k_\theta(X_C,Y_C)\end{aligned}$$

where Y_C is a random variable in the random interval $[1, X_C] \cup [X_C, 1]$ and

$$\begin{cases} h_\theta(X_C) = f_\theta(1) + f'_\theta(1)\,(X_C - 1) + \dfrac{f''_\theta(1)}{2}\,(X_C - 1)^2\,, \\ \\ k_\theta(X_C, Y_C) = \dfrac{f''_\theta(Y_C) - f''_\theta(1)}{2}\,(X_C - 1)^2\,. \end{cases}$$

With the latter expansion, we can then decompose $B_C = D_C + E_C$ where

$$D_C = \mathbb{E}\big[h_\theta(X_C)\mathbb{1}_{|X_C-1|<\eta}\big]\,, \quad E_C = \mathbb{E}\big[k_\theta(X_C, Y_C)\mathbb{1}_{|X_C-1|<\eta}\big]\,.$$

We then compute

$$D_C = \mathbb{E}[h_\theta(X_C)] - \mathbb{E}\big[h_\theta(X_C)\mathbb{1}_{|X_C-1|\geq\eta}\big] \tag{3.20}$$

with

$$\mathbb{E}[h_\theta(X_C)] = f_\theta(1) + \frac{f''_\theta(1)}{2C}$$

since $\mathbb{E}[X_C - 1] = 0$ and $\mathbb{E}\big[(X_C - 1)^2\big] = 1/C$. Besides, to deal with the term $\mathbb{E}\big[h_\theta(X_C)\mathbb{1}_{|X_C-1|\geq\eta}\big]$ in the right-hand side of (3.20), we use the Cauchy-Schwarz inequality to write

$$\big|\mathbb{E}\big[h_\theta(X_C)\mathbb{1}_{|X_C-1|\geq\eta}\big]\big| \leq \sqrt{\mathbb{E}[h_\theta(X_C)^2]}\sqrt{\mathbb{P}[|X_C - 1| \geq \eta]}$$

and note that $\mathbb{E}\big[h_\theta(X_C)^2\big] = O(1)$ for all $C > 1$ by Lemma 6 (ii); applying Lemma 6 (i) then eventually shows that $\mathbb{E}\big[h_\theta(X_C)\mathbb{1}_{|X_C-1|\geq\eta}\big]$ is $O(e^{-\frac{C}{2}\cdot\varphi(1+\eta)})$ which is, in particular, $o(1/C)$. At this stage, we therefore conclude from (3.20) and the latter discussion that

$$D_C = f_\theta(1) + \frac{f''_\theta(1)}{2C} + o\left(\frac{1}{C}\right). \tag{3.21}$$

Lastly, we show that the term E_C is $o(1/C)$. To this aim, it is sufficient to show (see [62], Theorem 13.7) that the sequence $W_C = C \cdot k_\theta(X_C, Y_C)$, $C > 1$, converges in probability to zero and that it is uniformly integrable:

• to prove the convergence in probability, note that since $X_C \to 1$ a.s. when $C \to \infty$ and $Y_C \in [1, X_C]$, then $Y_C \to 1$ a.s. It follows from the continuity of f''_θ in the interval $(1 - \eta, 1 + \eta)$ that $f''_\theta(1) - f''_\theta(Y_C) \to 0$ a.s. and, in particular, in probability. On the other hand, since $X_C = \widehat{T}_C/C$ is an average of C i.i.d. random variables with mean 1, the continuous mapping theorem for weak limits implies that $C(X_C - 1)^2$ converges in distribution (the limit distribution is χ^2 with parameter 1 but this specific limit has no importance for the present proof). Finally, since $\mathbb{1}\{|X_C - 1| < \eta\} \to 1$ a.s.,

Slutsky's theorem (Th. 11.4 in [32]) allows us to conclude that

$$W_C = \frac{f''_\theta(1) - f''_\theta(Y_C)}{2} \times C(X_C - 1)^2 \times \mathbb{1}\{|X_C - 1| < \eta\} \to 0$$

in distribution as $C \to \infty$ and thus in probability as well;

• to prove the uniform integrability of W_C, it suffices to show that

$$\sup_{C \geq 1} \mathbb{E}\left[W_C^2\right] < \infty \tag{3.22}$$

(see [62] Theorem 13.3). First note that, since f_θ is twice continuous differentiable, we have

$$\left| \frac{f''_\theta(1) - f''_\theta(Y_C)}{2} \mathbb{1}_{|X_C - 1| < \eta} \right| \leq K$$

for any $C > 1$ and for some constant K depending on η only. Secondly, by Lemma 6 (ii), we further have $\mathbb{E}\left[C^2 (X_C - 1)^4\right] = C^2 \times O(C^{-2}) = O(1)$. We finally conclude that $\mathbb{E}\left[W_C^2\right] < K^2 \times O(1) < \infty$ which proves the claimed property (3.22).

Finally gathering $\mathbb{E}[\mu_C] = A_C + B_C = A_C + D_C + E_C$ with $A_C = o(1/C)$, $E_C = o(1/C)$ and D_C expanded in (3.21), we thus have proved that

$$\mathbb{E}[\mu_C] = f_\theta(1) + \frac{f''_\theta(1)}{2C} + o\left(\frac{1}{C}\right) \tag{3.23}$$

as $C \to \infty$. To conclude the proof, we now express function f_θ and its derivatives at 1 in terms of the function m and its derivatives at t_θ; by implicit differentiation, we calculate

$$f'_\theta(z) = \frac{m'(t_{\theta z})}{m(t_{\theta z})}\theta, \quad f''_\theta(z) = \frac{\theta^2}{m(t_{\theta z})^2}\left(m''(t_{\theta z}) - \frac{m'(t_{\theta z})^2}{m(t_{\theta z})}\right);$$

the values of f'_θ and f''_θ at $z = 1$ consequently follow and replacing them into (3.23), we finally prove the expansion (3.13), as claimed.

Proof of Proposition 10

Differentiating (3.7) under the integral sign, with $\lambda(u)$ expressed by (3.14), readily gives formulas (3.15) after using the variable change $u \mapsto u/L$. The validity of these formulas can then be simply proved by showing that these integrals for m' and m'' are finite.

Given $t > 0$ and L, define $u^* = u^*(t, L) = \inf\{u : f(u) > f(u+t/L)\}$, so that $f(u) \leq f(u+t/L)$ for $u \leq u^*$ and $f(u) > f(u+t/L)$ for $u > u^*$. The existence of u^* is ensured from the unimodality of f, and we have $u^* = 0$ if and only if f is non-increasing. Finally, define $\tilde{u} = \inf\{u : f(u) = \max f\}$

(see Figure 3.5 for an schematic view of these definitions).

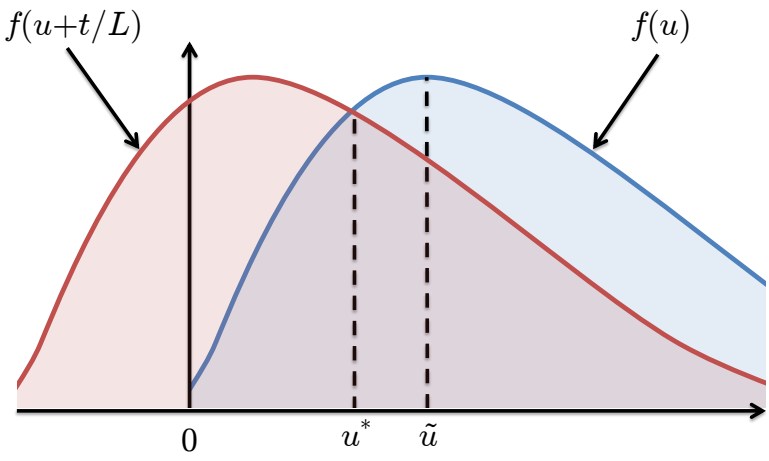


Figure 3.5: *Schema for unimodal f*

Since f is differentiable and unimodal, it is quasi-concave (see [19], Lemma 2.4.1.), that is, for any $0 \le \eta \le 1$, we have $f(\eta u_1 + (1-\eta)u_2) \ge f(u_1) \wedge f(u_2)$ for $u_1, u_2 \ge 0$. As a consequence, for any $t > 0$, the area under the graph of f in the interval $[u, u + t/L]$ can be bounded below by

$$F(u + t/L) - F(u) \ge \begin{cases} f(u) \cdot t/L, & u \le u^*, \\ f(u + t/L) \cdot t/L, & u > u^*. \end{cases} \tag{3.24}$$

We now divide the integrals in (3.15) into their contributions from intervals $[0, u^*]$ and $[u^*, \infty)$, respectively, and bound them separately. For the first derivative $m(t)$, using lower bounds (3.24) we obtain

$$\begin{aligned} |m'(t)| &\le \mathbb{E}\left[RL \int_0^{u^*} f(u + t/L) R f(u) e^{-Rf(u)t} \, \mathrm{d}u\right] \\ &\quad + \mathbb{E}\left[RL \int_{u^*}^{\infty} f(u) R f(u + t/L) e^{-Rf(u+t/L)t} \, \mathrm{d}u\right] \le \frac{2}{et} \mathbb{E}[RL] \end{aligned}$$

where the last inequality is justified by the bound $xe^{-ax} \le 1/ae$ for any fixed $a > 0$, and the fact that $\int f = 1$.

For the second derivative $m''(t)$, we introduce integrals

$$A_1(t) = \mathbb{E}\left[RL \int_0^\infty R^2 f(u) f(u+t/L)^2 e^{-RL(F(u+t/L)-F(u))\,\mathrm{d}u}\right],$$
$$A_2(t) = \mathbb{E}\left[R \int_0^\infty Rf(u) f'(u+t/L) e^{-RL(F(u+t/L)-F(u))\,\mathrm{d}u}\right]$$

so that $|m''(t)| \le |A_1(t)| + |A_2(t)|$. For $A_1(t)$, we have

$$\begin{aligned}|A_1(t)| &\le \mathbb{E}\left[RL \int_0^{u^*} f(u+t/L)^2 f(u) R^2 e^{-Rf(u)t}\,\mathrm{d}u\right] \\ &\quad + \mathbb{E}\left[RL \int_{u^*}^\infty f(u) R^2 f(u+t/L)^2 e^{-Rf(u+t/L)t}\,\mathrm{d}u\right] \\ &\le \mathbb{E}\left[\frac{4RL}{e^2 t^2 f(0)} \int f^2\right] + \mathbb{E}\left[\frac{RL}{et}\right] \le \frac{1}{et}\left(1 + \frac{4}{f(0)et}\int f^2\right)\mathbb{E}[RL] < \infty\end{aligned}$$

where the last inequality follows from the bounds $xe^{-ax} \le 1/ae$, $x^2 e^{-ax} \le 4/a^2 e^2$ for any fixed $a > 0$, and the fact that $0 < f(0) \le f(u) \le f(u+t/L)$ for $u \in [0, u^*]$. Regarding $A_2(t)$, we have

$$\begin{aligned}|A_2(t)| &\le \mathbb{E}\left[R \int_0^{u^*} Rf(u)|f'(u+t/L)| e^{-Rf(u)t}\,\mathrm{d}u\right] \\ &\quad + \mathbb{E}\left[R \int_{u^*}^\infty Rf(u)|f'(u+t/L)| e^{-Rf(u+t/L)t}\,\mathrm{d}u\right] \\ &= B_1(t) + B_2(t).\end{aligned}$$

Using again $xe^{-ax} \le 1/ae$, we have

$$B_1(t) \le \frac{\mathbb{E}[R]}{et} \int |f'| < \infty.$$

Finally, to deal with $B_2(t)$ we note that $f'(u+t/L) \le 0$ for $u \in [u^*, \infty)$ and thus $|f'(u+t/L)| = -f'(u+t/L)$. We then use an integration by parts to obtain

$$\begin{aligned}B_2(t) &= -\frac{1}{t}\mathbb{E}\left[R\left(\left[-e^{-Rf(u+t/L)t} f(u)\right]_{u=u^*}^\infty + \int_{u^*}^\infty f'(u) e^{-Rf(u+t/L)t}\,\mathrm{d}u\right)\right] \\ &= -\frac{1}{t}\mathbb{E}\left[Rf(u^*) e^{-Rf(u^*+t/L)}\right] \\ &\quad - \mathbb{E}\left[R\int_{u^*}^{\tilde{u}} f'(u) e^{-Rf(u+t/L)t}\,\mathrm{d}u\right] - \mathbb{E}\left[R\int_{\tilde{u}}^\infty f'(u) e^{-Rf(u+t/L)t}\,\mathrm{d}u\right].\end{aligned}$$

The first term in the latter expression is trivially negative; the second is also negative since f is

non-decreasing in $[0, \tilde{u})$. As a consequence both terms can be ignored to obtain

$$B_2(t) \leq \frac{1}{t}\mathbb{E}\left[R\int_{\tilde{u}}^{\infty}|f'(u)|e^{-Rf(u+t/L)t}\,\mathrm{d}u\right] \leq \frac{\mathbb{E}[R]}{t}\int_0^{\infty}|f'(u)|\mathrm{d}u < \infty$$

where the last inequality again follows from $xe^{-ax} \leq 1/ae$, thus concluding the proof.

Chapter 4

Parameter Estimation

We now devote our efforts to the topic of parameter estimation for performance modeling. Already in Chapter 2, we have used a parameter fitting procedure for the validation of the Box model. This procedure, although effective, is ad-hoc and has the inconvenience of being defined only for the sub-sample of objects with at least two requests. The main reason for the difficulties in this estimation is that, when we represent a traffic trace via the Box model, the popularity and the lifespan are unobserved random variables.

One methodology that allows to deal with the latter problem is provided by Maximum Likelihood (ML) estimations. The main objective of this chapter is to propose a ML-based parameter fitting method in the simpler case of the IRM model. However, due combinatorial explosion of its likelihood function, the ML method is not suitable to the IRM model. Therefore, we propose a modified version of IRM we call IRM-Mixed (IRM-M) in which the document popularity is modeled by an i.i.d. sample from a probability distribution. In consequence the individual document request sequences are mixed Poisson processes and thus we ca regard IRM-M as an intermediate model between IRM and the Box model.

While the IRM-M models the same localities as IRM, the additional randomness layer renders its likelihood function tractable. This enables us to apply the ML method for the estimation of its popularity distribution. At the same time, the ML method allows us to seamlessly solve the issue of unobserved documents and to use the whole dataset without using ad-hoc tinkering as we did in Chapter 2.

We remark that solving this problem has applications other than performance evaluation. To highlight this fact, we show in Figure 4.1 an expanded version of our performance evaluation workflow. For example, with the estimated parameters in hand, we could resample synthetic traces that are statistically similar to the original one. These traces could be used to realistically simulate the behavior of other systems receiving such a traffic.

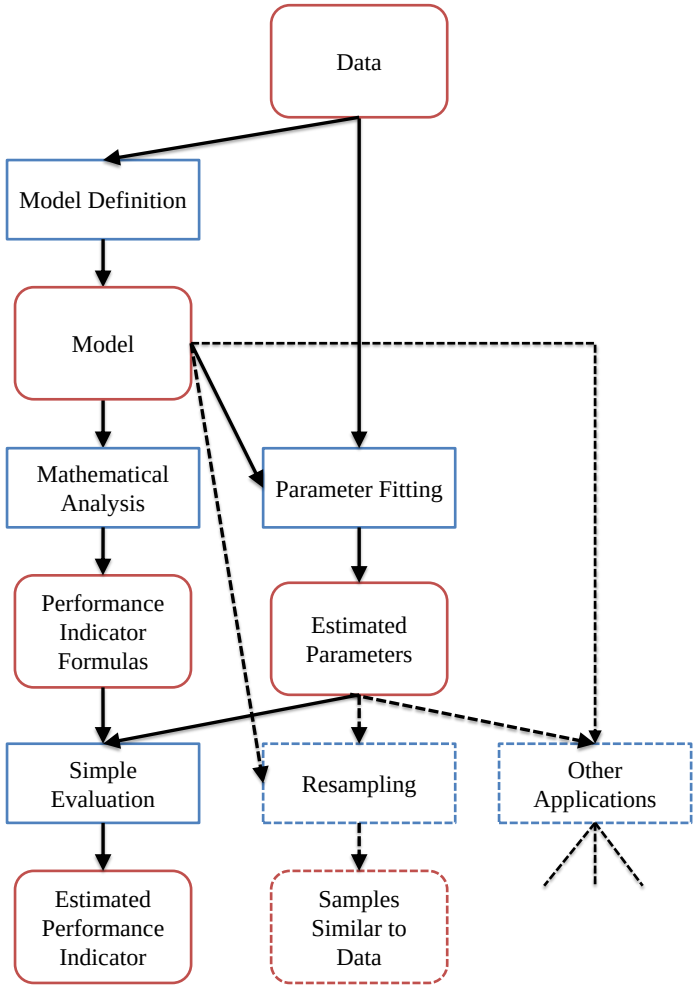


Figure 4.1: *Expanded performance evaluation workflow with a fixed model.*

We start this chapter by briefly describing two additional synthetic datasets we use in this chapter and by making explicit the problem to solve.

4.1 Additional Datasets

In addition to `#yt` and `#vod` datasets discussed in previous chapters, we add two synthetic datasets called `#prt` and `#delta`. These datasets allow us to highlight in a clearer way some of our findings and, more importantly, to validate the results with controlled experiments where ground-truth is available. The set `#prt` (resp. `#delta`) is generated by first drawing 10 000 000 (resp. 100 000) random samples

with Pareto $(1.6, 0.1)$ (resp. Dirac delta at 4.0) distribution representing the popularity (see Section 4.3 for a model description). The number of requests for each document is then drawn according to the Poisson distribution with mean equal to the document popularity. After discarding the documents with zero request, this results into $2\,600\,000$ (resp. $400\,000$) requests to $1\,900\,000$ (resp. $98\,000$) documents.

4.2 Problem Definition

Recall that, in the case of LRU cache performance evaluation with IRM traffic, users request documents among a catalog of K documents. Under IRM, the sequence of requests for document $1 \leq k \leq K$ is a Poisson process with rate r_k, where r_k is proportional to the popularity of document k; all such processes are mutually independent and their superposition build up the total request process. In this model, the number N_k of requests for document k in a time window W is an independent Poisson random variable $\mathcal{P}(r_k W)$ of mean $r_k W$. Up to a time normalization, we assume in the following that $W = 1$.

Assume now that an observer has access to a traffic trace. In the case of IRM, a sufficient statistic of the request process are the request counts $n_1, n_2, \ldots, n_{K_0}$ for all observed document, where K_0 is the number of observed documents in the sample. Following the point of view of an Internet Service Provider (ISP), we here assume that objects with zero request *are not observable* in the sample. Our main objective is to solve the following problem:

Problem Statement (First Version)
Obtain a popularity distribution estimation such that the request flow predicted by the model using these parameters represents the data at best

A simple solution, henceforth called the *naive method*, is to estimate the popularity of a document by its request count and the catalog size by the number of observed objects, that is:

$$\widehat{K}^{\text{nv}} = K_0 \quad \text{and} \quad \widehat{r}_k^{\text{nv}} = n_k \quad \text{for} \quad 1 \leq k \leq \widehat{K}^{\text{nv}}$$

We identify two problems at this stage. First, since the trace is zero-censored, with high probability the observed number of documents K_0 is strictly smaller than the catalog size K. Second, each document popularity r_k is estimated by a single sample n_k of the random count N_k. This last limitation is well illustrated in the case of the `#delta` dataset. By definition, the ground-truth popularities are $r_k = 4$. In the dataset, however, the counts of document requests are Poisson random variables with mean 4, hence $\widehat{r}_k^{\text{nv}} = \mathcal{P}(4)$ and the naive estimation "dilutes" the mass of popularities over the set of positive integers. In Figure 4.2, we show the impact of these limitations for the hit ratio estimation, based on the `#prt` trace. The first curve is our ground-truth. It is obtained via simulation of a LRU

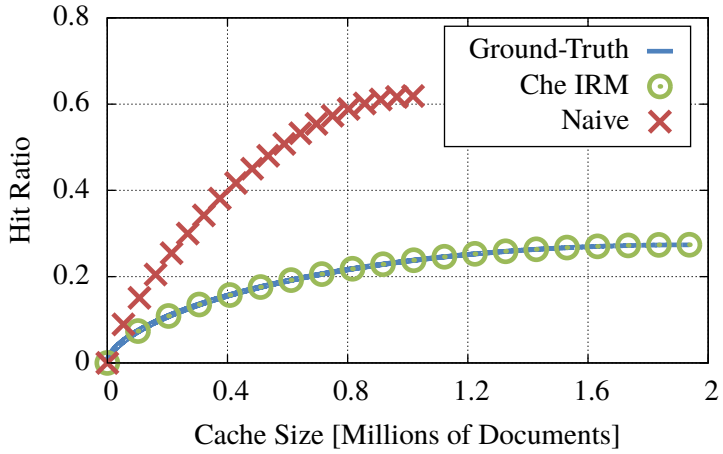


Figure 4.2: *Hit ratio of a cache fed by* `#prt` *trace: Ground-truth and prediction by the naive estimation. The cache size is normalized with respect to that of the ground-truth.*

cache starting empty; the cache is fed by the traffic trace that is randomly shuffled to enforce the IRM assumption. The second curve is the prediction of the IRM model, when fed by the real popularities in the trace (see Section 6.1 for a quick derivation of the transient hit ratio for the IRM). As expected, it perfectly fits the ground-truth. The third curve shows the results obtained by the IRM model when fed by the parameters $\widehat{K}^{\text{nv}}$ and $\widehat{r}_k^{\text{nv}}, 1 \leq k \leq \widehat{K}^{\text{nv}}$, from the naive estimation. The hit ratio curves are seen to clearly differ, and the naive method proves inaccurate for estimating document popularities when fitting a performance model.

In the absence of any prior knowledge about the popularity distribution, the only available data for the estimation of each document popularity is a single request count, which limits the accuracy of this approach. To overcome this lack of information, we thus aim at jointly estimating the set of popularities, from the joint set of request counts. The latter approach allows us to use all the information contained in the joint Poisson distribution rather than just the mean.

We now make more precise our problem with the previously introduced notations:

Problem Statement (Second Version)
Given the measured request counts $\{n_1, n_2, \ldots, n_{K_0}\}$*, determine the parameters* $\widehat{K}$ *and* $\widehat{r}_1, \widehat{r}_2, \ldots, \widehat{r}_{\widehat{K}}$ *so that the set of random variables* $\{N_1, N_2, \ldots, N_{\widehat{K}}\}$*, where* $N_k = \mathcal{P}(\widehat{r}_k)$ *for* $1 \leq k \leq \widehat{K}$*, is the "closest" to* $\{n_1, n_2, \ldots, n_{K_0}, 0, \ldots, 0\}$*, with* $\widehat{K} - K_0$ *zeros at the tail.*

4.3 Maximum Likelihood Estimation

In this section, we show how to solve the latter inverse problem via the Maximum Likelihood method.

In the IRM setting, the parameters $(r_1, r_2, \ldots, r_K, K)$ are not ordered, and thus every request count could correspond to any of the popularities. The likelihood given observations $n = (n_1, n_2, \ldots, n_K)$ thus runs through every permutation σ of size K. Specifically, the likelihood $\mathcal{L}$ is given by

$$\mathcal{L}(r_1, r_2, \ldots, r_K, K;\, n) = \frac{1}{K!} \sum_{\sigma} \left(\prod_{j=1}^{K_0} \frac{e^{-r_{\sigma(j)}} r_{\sigma(j)}^{n_j}}{n_j!} \times \prod_{j=K_0+1}^{K} e^{-r_{\sigma(j)}} \right).$$

The combinatorial explosion incurred in the evaluation of $\mathcal{L}$ for large catalog size K makes the ML method intractable for the IRM model. We thus propose in the following a slightly modified model, which is simultaneously tractable for ML estimations and simple to analyze.

IRM Mixture Model (IRM-M)

In order to succinctly describe the popularity parameters $r_1, r_2, \ldots, r_K$ and to ease their estimation, we slightly modify the IRM model by considering them as random variables. Specifically, we now model the popularity by an i.i.d. sample $R_1, R_2, \ldots, R_K$ from an unknown *mixing distribution* with density g. Given the value of R_k, the request process to the k^{th} document remains a Poisson process with intensity R_k, and thus the counts of each document follow a mixed Poisson distribution with some mixing distribution g. In particular, the number of requests N for any document satisfies

$$\mathbb{P}[N = j] = \mathbb{E}_g\left[\frac{e^{-R}R^j}{j!}\right], = \int_0^\infty \frac{e^{-x}x^j}{j!} g(x)\,\mathrm{d}x \tag{4.1}$$

$$\mathbb{P}[N > 0] = \mathbb{E}_g\left[1 - e^{-R}\right] = \int_0^\infty (1 - e^{-x}) g(x)\,\mathrm{d}x \tag{4.2}$$

for $j \in \mathbb{N}$, where the operator $\mathbb{E}_g[\cdot]$ represents the expectation a under the mixing distribution g.

ML estimation on IRM-M

By modifying the model, we have changed the problem of estimating the static parameters $r_1, r_2, \ldots, r_K$, to that of estimating the mixing distribution g.

Problem Statement (IRM-M)
Given the measured request counts $\{n_1, n_2, \ldots, n_{K_0}\}$*, determine the catalog size* $\widehat{K}$ *and the mixing density* $\widehat{g}$ *such that an i.i.d. mixed Poisson sample* $\{N_1, N_2, \ldots, N_{\widehat{K}}\}$ *is the "closest" to the set* $\{n_1, n_2, \ldots, n_{K_0}, 0, \ldots, 0\}$*, with* $\widehat{K} - K_0$ *zeros at the tail.*

We now show how this problem can be solved via a ML method. Let $J = \max_{k=1}^{K_0}\{n_k\}$ be the maximum number of requests over all documents, and let

$$\mu_j = \frac{1}{K_0}\sum_{k=1}^{K_0} \mathbb{1}\{n_k = j\}$$

be the proportion of documents with j requests, $1 \leq j \leq J$. Using (4.1) and (4.2), the log-likelihood $\ell(g;\, \mu)$ of the popularity distribution g for the observations $\mu = (\mu_j)_{j\geq 1}$ reads

$$\begin{aligned}\ell(g;\, \mu) &= \sum_{j=1}^{J} \mu_j \log \mathbb{P}[N = j \mid N > 0] \\ &= \sum_{j=1}^{J} \mu_j \log \mathbb{E}_g\left[\frac{e^{-R}R^j}{j!}\right] - \log \mathbb{E}_g\left[1 - e^{-R}\right].\end{aligned}$$

We remark that, in this setting, the catalog size K is decoupled from the popularity distribution. Thus, we can first obtain an estimator $\widehat{g}$ of the mixing distribution g, and then approximate K by

$$\widehat{K}^{\mathrm{ml}} = \frac{K_0}{\mathbb{E}_{\widehat{g}}[1 - e^{-R}]} \tag{4.3}$$

which is asymptotically close to the ML estimator.

We now proceed with the detailed form of the likelihood function for the *parametric* and *non-parametric* estimation procedures. In both approaches, we numerically solve the problems with a generic non-linear optimization solver in MATLAB based on an interior point algorithm. Our code is freely available online.[1] We discuss the use of specialized algorithms in Section 4.5.

Parametric Estimation

In this setting, we determine the mixing distribution g within a parametric family of density functions whose choice relies on an a-priori knowledge. The computation of the ML estimator obviously depends on this choice, and due to space restriction, we here limit ourselves to the two-parameter Pareto family

[1] Code : http://www.olmos.cl/code/mixed_poisson.tgz

with densities

$$g(x) = \frac{\alpha x_m^\alpha}{x^{\alpha+1}}$$

for $x > x_m$, with α, x_m the shape and scale parameters, respectively. The log-likelihood function $\ell = \ell(\alpha, x_m; \mu)$ then reads

$$\ell = \sum_{j=1}^{J} \mu_j \log \frac{\Gamma(j-\alpha, x_m)}{j!} - \log\left(\alpha x_m^\alpha - \Gamma(-\alpha, x_m)\right)$$

where Γ is the incomplete Gamma function.

Non-Parametric Family

In the absence of a-priori knowledge about the distribution g, the non-parametric (NP) approach provides a method to obtain an estimator. In this setting, we determine a discrete distribution g of the form $\mathbb{P}[R = x_i] = \theta_i$ for $1 < i < I$. The log-likelihood correspondingly reads

$$\ell(\theta; \mu) = \sum_{j=1}^{J} \mu_j \log \sum_{i=1}^{I} \theta_i \frac{e^{-x_i} x_i^j}{j!} - \log \sum_{i=1}^{I} \theta_i (1 - e^{-x_i}).$$

Hit Probability Analysis

As detailed in Appendix 6.1, the IRM-M model proves to be tractable for evaluating the performance of an LRU cache. In particular, the Che approximation is easily adapted to the IRM-M case; furthermore, we are able to derive formulas for the transient analysis of the hit ratio, when starting from an empty cache.

4.4 Numerical Evaluation

The accuracy of the parameter estimation can be evaluated at three different levels, as expressed by the following questions:

(1) Is the estimated popularity distribution close to the actual one?

(2) Is the request flow predicted by the model statistically similar to the actual one?

(3) Is the hit probability of the fitted model accurately predicted?

As in Chapter 2, we assess the precision of a curve estimate by computing the so-called *mean absolute percentage error* (MAPE) defined in Equation (2.1).

Estimation of popularity distribution

First, we start with the most general question, that is, the estimation of the mixing distribution.

By means of the NP method, we obtain an estimate $\widehat{g}^{\text{np}}$ of the popularity density by applying the NP method, using a support with 0.01 as lower bound, exponentially increasing spacings and an upper bound slightly larger than the maximum of observed requests (e.g., $2\,400$ for `#prt` and 16 for `#delta`). The naive fitting corresponds to the empirical measure of the request counts, that is, the mixture of Dirac measures

$$\frac{1}{K_0}\sum_{k=1}^{K_0}\delta_{n_k}(\cdot).$$

We observe in Figure 4.3 the NP estimator of the mixing distribution for the `#delta` and `#prt` datasets. In the `#delta` case, the ground-truth is a Dirac measure at $R = 4$, and the naive method fails at correctly estimating its shape, whereas the ML estimator concentrates its mass around the value $R = 4$. In the `#prt` case, as expected, the estimated distribution is irregular, tending to accumulate mass at certain points (see Section 4.5 for possible regularization solutions). The peaks, nevertheless, capture the power law trend, as reflected by the good estimation quality of the mixture distribution. In contrast, the naive method fails at correctly estimating both the trend of distribution body and its tail.

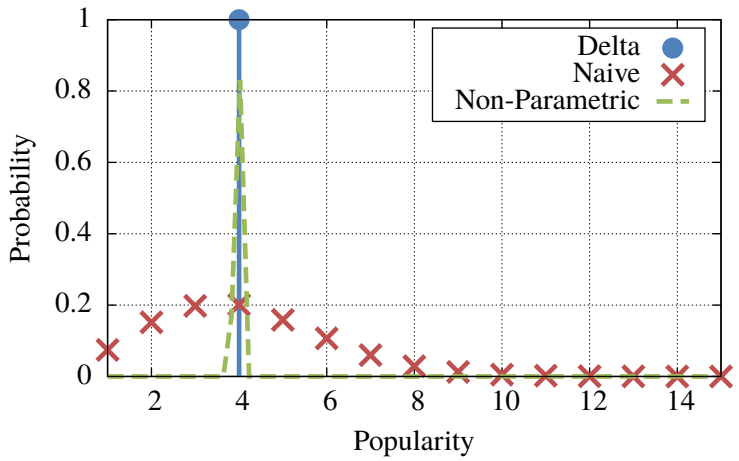
Using Equation (4.3), we also calculate the catalog size, giving $\widehat{K} \approx 11\,600\,000$ (resp. $105\,278$) for the `#prt` (resp. `#delta`) case. This represents a relative error of 11.6% and 5.2%, respectively. Following Equation (4.3), it shows that estimating the probability that a document receives no request for the duration of the trace, based on the very same trace, is a difficult task. As a consequence, this error is not negligible. It is, however, smaller, and even more significantly in the `#prt` case, than the relative error of the naive method (recall that $\widehat{K}^{\text{nv}} = K_0 = 1\,900\,000$ and $\widehat{K}^{\text{nv}} = 92\,046$ for the `#prt` and `#delta` traces, respectively).

When some a priori knowledge about the distribution shape is available, the estimates can be improved via the parametric approach. In the `#prt` case, the resulting Pareto fit gives the estimates $\widehat{\alpha} = 1.597$ and $\widehat{x}_m = 0.099$ that are very close to the original parameters $\alpha = 1.6$ and $x_m = 0.1$. We compare these results to that of the "log-log" approach, which consists in estimating the tail index by fitting a least square approximation to the log-log rank-frequency plot, as shown in Figure 4.4. The rank frequency plot roughly decays as $1/\alpha$. Using the first $20\,000$ objects to compute the regression, the estimation gives 1.704, which is worse than the ML estimate.

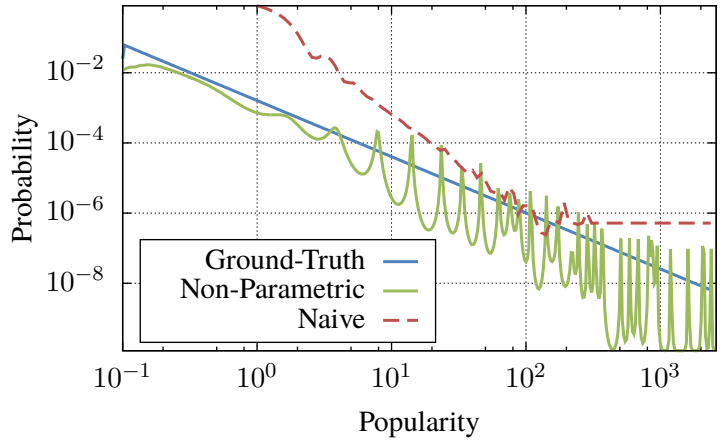
Request flow estimation

In this section, we specify the discussion by estimating the zero-censored request count distribution (or mixture distribution in statistical terms) $\mathbb{P}[N = j \mid N > 0]$, $j \geq 1$.

For the naive approach, we use the results of the previous section to generate an IRM trace using



(a) `#delta` trace



(b) `#prt` trace

Figure 4.3: *Mixing distribution obtained via the non-parametric methods.*

the estimated parameters. We then count the mixture distribution. The experiment is repeated $50\,000$ times, with a coefficient of variation lower that 10^{-4} for all points of the distribution. As regards the ML approach, using the $\widehat{g}^{\text{np}}$ density, we numerically compute the associated zero-censored request distribution using Equation (4.1).

In Figure 4.5, we show the resulting zero-censored request distribution estimated by each method. For comparison, we include the real mixture distribution for the `#prt` dataset, which can be calculated explicitly. For the `#yt` and `#vod` datasets, we show instead the observed request distribution.

We observe two issues in the naive approach, that are not present in the maximum likelihood estimation:

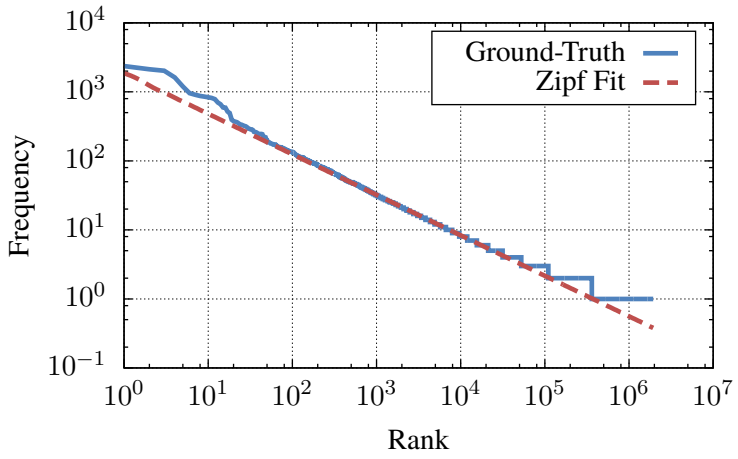


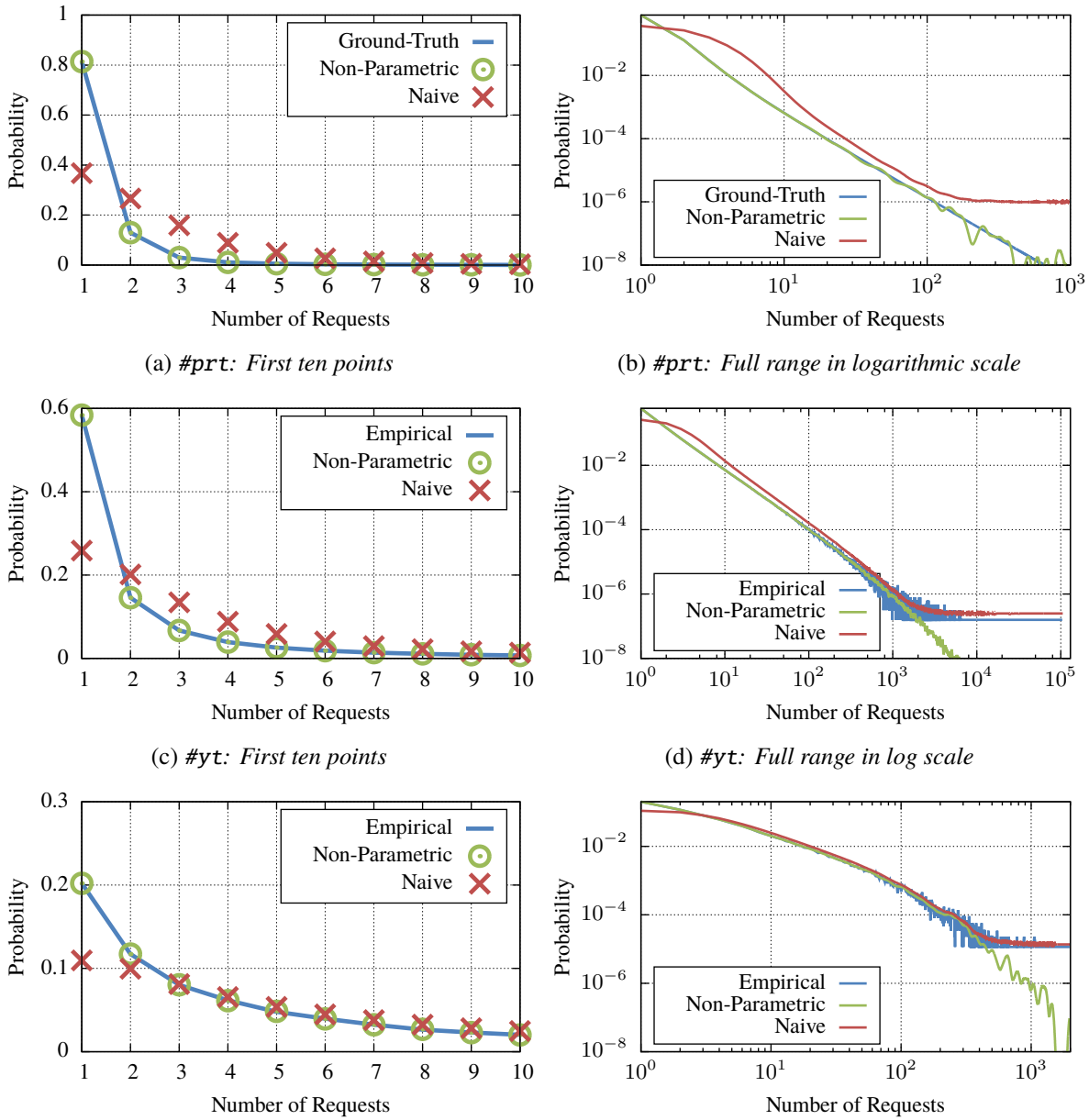
Figure 4.4: *Rank frequency distribution for the* `#prt` *trace*

- first, at the head of the distribution, where most of the mass is concentrated, large estimation errors are produced by the naive approach. Such errors produce a mass shift towards the tail of the distribution. On the contrary, the NP estimation matches perfectly the head of the distribution;
- second, the naive method over-fits the tail of the distribution. We observe in Figure 4.5b that the naive estimate shows a "horizontal branch" at the tail, and differs significantly from the ground-truth that is approximately a straight "diagonal" line. This horizontal branch is in fact a few isolated masses, though they look as a line on the figure. The naive estimation therefore concentrates the mass of the ground-truth distribution on a few points. On the other side, the ML estimation correctly estimates the trend of the distribution at all scales, though noise inaccuracies appear at the tail. This is quantified by the MAPE of 1.67 for the ML estimation, whereas the naive method leads to a MAPE of 668, for the full range distribution. As regards the `#yt` and `#vod` cases in Figures 4.5d and 4.5f, we similarly observe the same horizontal branch at the tail for the naive distribution. In the absence of available ground-truth, we do not compute the MAPE, but the similarity of behavior hints that the ML method also performs better on these traces.

Hit Probability Estimation

We finally compare the hit probabilities predicted by the IRM-M model with popularity distributions fitted using the naive and the ML methods, both for the `#prt` and `#yt` traces.

Figure 4.6 shows the obtained hit probability curve in each case. In the case of the `#yt` trace, in order to obtain a request sequence modelable by IRM-M, we applied the global randomization procedure as in Section 2.2. The ground-truth curve is then obtained by simulating a cache with each trace. The Naive



(a) `#prt`: *First ten points*

(b) `#prt`: *Full range in logarithmic scale*

(c) `#yt`: *First ten points*

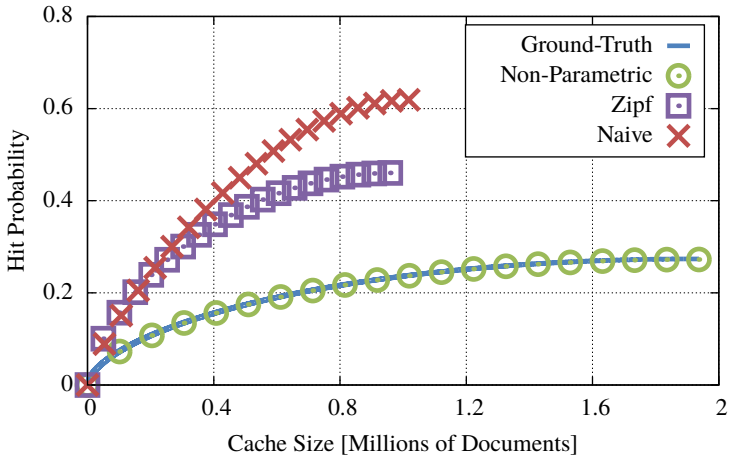
(d) `#yt`: *Full range in log scale*

(e) `#vod`: *First ten points*

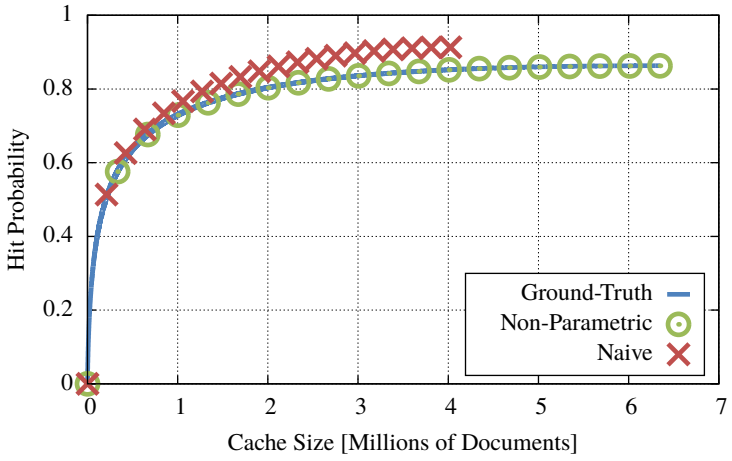
(f) `#vod`: *Full range in log scale*

Figure 4.5: *Censored Mixture distribution estimations obtained with the non-parametric method*

(resp. NP) curves are obtained when using formula (6.4) (resp. (6.7)) with the parameters obtained by the naive (resp. NP) method. Finally, the Zipf curve, for the `#prt` trace, corresponds to the hit probability prediction when using the "log-log" parametric fitting method detailed in Section 4.4 in conjunction with formula (6.4).



(a) *#prt trace*



(b) *#yt trace*

Figure 4.6: *Hit probability estimations*

The naive approach leads to small inaccuracy for the #yt trace and large errors for the #prt trace, with respective MAPE of 0.06 and 1.44. This difference in estimation accuracy can be explained by the variability of the random variable N. Indeed, in the #yt dataset, documents receive an average of 7.3 requests per document, whereas this average decreases to 1.4 in the #prt trace. It follows that the coefficient of variation of the request count distribution is greater in the #prt trace than in the #yt trace. As expected, the inaccuracy of the naive method is greater for the former than for the latter. Note also that from an operational point of view, the focus is on the miss probability, which determines the dimensioning requirements upstream of the cache. The inaccuracy of the naive hit probability prediction for the #yt dataset becomes relatively significant in this context. As shown by the Zipf curve,

the knowledge of a relevant parametric family allows us to improve the hit probability estimation. The error, however, remains significant with a MAPE of 0.96. In contrast, the non-parametric ML curves match perfectly the original ones, as shown by the MAPE of 0.002 for the `#yt` trace and 0.005 for the `#prt` trace. We conclude that, as regards hit probability, our estimation method accurately estimates the model parameters. In contrast, in the Zipf case, a seemingly small error of 0.1 in the estimation of the tail exponent leads to a significant error in the hit probability estimation.

4.5 Discussion and Conclusion

Other Applications and Extensions

Since our methodology requires only the statistics about the number of requests per document, the presented estimation method for content popularity can be readily applied in use-cases other than caching performance. For example, the estimations can be used for dimensioning the bandwidth in the access network for VoD or TV multicast services or even predicting the demand for content in marketing studies.

Additionally, the wide applicability of the ML estimators makes our method a viable option for other traffic models. In particular, our framework can be extended to to renewal [23, 7] and the cluster processes we have analyzed in previous chapters. In these cases new challenges arises, due the reformulation of the ML method. For example, in the Box model, the randomized parameter is not univariate, but multivariate or can even an stochastic process [23]. Another factor to consider is time censure, due the greater impact of the time variable in stochastic models other than IRM-M.

Maximization techniques

The main current limitation of our maximization approach is that the estimated mixing density exhibits a lot of peaks, which is consistent with the results of Lindsay [43]. This might be a problem when one aims at understanding the nature of the popularity distribution.

A possible solution to enforce smoothness in the mixing density estimation is to introduce a penalization for the irregularities. Classical candidates for such a penalization are the L^2-penalization or a logarithmic penalization

$$P(\theta) = \sum_{i=1}^{I} (\log \theta_{i+1} - \log \theta_i) \frac{\theta_{i+1} - \theta_i}{x_{i+1} - x_i}.$$

One then maximizes $\ell(\theta; \mu) - \rho P(\theta)$, where ρ represents the trade-off factor between fitness and smoothness. Regularization here comes at the price of choosing the right penalization function P and

the right value of ρ and in our case, the results have been satisfactory only for concentrated mixing distributions.

Another possibility is to exploit the fact that the peaks conserve the overall trend of the distribution. We thus extract the peak locations. A second ML optimization is then performed using these peak locations as the new support. Though non-standard, this gives satisfactory results for the `#prt` dataset as shown in Figure 4.7.

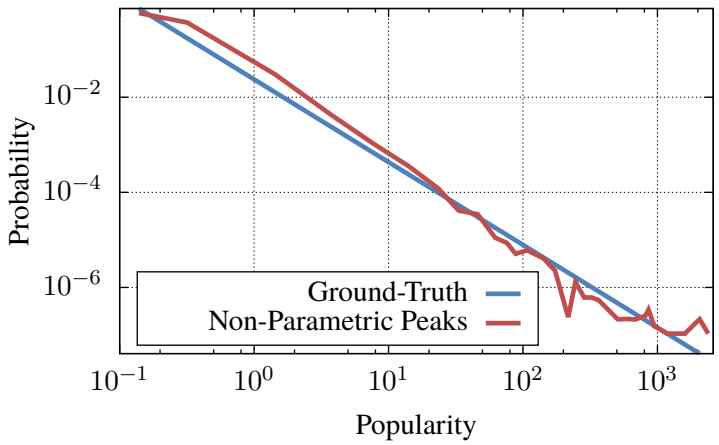


Figure 4.7: *Estimation of the mixing distribution by the peak selection method*

Summary of results

In this chapter, we have presented and solved the inverse problem that consists in estimating from a trace the popularity parameters for the performance evaluation of an LRU cache under the IRM-M request model. A key point in our approach is that we consider the probability that a document receives a given number of requests, rather than the probability that a request is directed to a given document. This representation is consistent with recently developed caching models (see Chapter 2 and [59, 23]). Moreover, it allows us to avoid the fitting of a rank-frequency plot, which is in essence an order statistic and exhibits over-fitting. Our second contribution on the modeling aspects is that we consider popularities as random variables, rather than parameters, leading to a mixture model tractable via ML methods. We have illustrated our method in the case of cache performance evaluation but our framework is applicable and extensible to other settings.

The inverse problem stems from the random nature of the requests count N for a given document. In particular, a traffic trace contains a single sample of these requests counts. The accuracy of any method that aims at fitting independently the popularity of each document is therefore limited by the inherent variability of the random variable N. The importance of using a sound methodology correspondingly

increases when the variability of the request counts is large, which is typically the case when N is small.

Determining the parameters of the model allows one to use the performance for diverse objectives, including the dimensioning of operational networks or the design of new mechanisms. More importantly, in contrast with simulation-based analysis, it enables one to more easily explore *what-if* scenarios, by keeping some parameters at their current value and modifying others to reflect future or possible changes.

Chapter 5

Conclusion & Perspectives

In this dissertation, we have proposed and implemented a framework to evaluate the performance of a LRU cache using mathematical modeling. This framework has at its core the usual mathematical analysis pertaining to performance evaluation, and extends it by incorporating a model selection step and a parameter fitting step. Starting from actual data traces, these three stages together provide a mathematical model for the traces that enables us to accurately estimate the hit probability of a LRU cache fed with such a traffic.

In Chapter 2, we have applied the "semi-experimental" method to the traces and subsequently proposed a parsimonious traffic model for our evaluation problem. The results of the semi-experiments have shown the relevant time correlations for LRU caching when the analysis is carried over a long period of time. These insights allowed us to propose the "Box Model" to represent the arrival sequences in the traces. This model is a Poisson cluster process in which the intensity of each cluster is a "box" function, and features a document catalog evolving in time.

In Chapter 3, we rigorously obtained an asymptotic estimation of the hit probability of a LRU cache for a general class of Poisson cluster processes that includes the "Box Model" as a particular case. We used results from Palm Theory to set up a probability space whereby a document can be tagged and analyzed independently from the rest of the system. This setup allows us to obtain an integral expression for the average number of misses that decouples the respective contributions of the tagged document and the rest of the process. Finally, by scaling the system, we obtained an asymptotic expansion for the hit probability with a large cache size C, justifying and quantifying the error of the widely used "Che Approximation".

In Chapter 4, we proposed a Maximum Likelihood method to estimate the popularity parameter of the IRM-M model. This method allows us to seamlessly handle the zero-censoring problem found in traffic traces. We show that although the hidden popularity distribution is difficult to be accurately

estimated, the method gives good results for the distribution of the number of requests and the theoretical performance of a LRU cache.

Perspectives

We now review some potential research directions to continue this work.

1) Mathematical Analysis

1.a) An immediate extension of our study, is to propose a more realistic model of the system by taking into account document sizes. These sizes and the cache size C should be measured for instance in bits, packets, or by a continuous value in $\mathbb{R}^+$. The document sizes can be incorporated as additional marks to the cluster point process. In this case, the process X defining the canonical exit time becomes a compound inhomogeneous Poisson process, summing up these file sizes. The exit time to consider for a canonical document of size S is then the first passage time of X strictly above the level $C - S$.
1.b) Our mathematical framework could be adapted to analyze other caching policies satisfying they property that the replacement algorithm for the canonical document depends only on the rest of the document request process. Examples of such caching policies found in the literature are RANDOM, which evicts a uniformly chosen document when adding a new document to the cache, and FIFO, which works as LRU except that it does not move a requested document that is already in the cache to the front of it. Such alternative policies may be relevant to ICN architectures: their simplicity may help to cope with the high line rates of in-network elements compensating their lower hit probability [28]. In order to analyze this case, the miss events for this policies are expected to be more intricate to analyze since they depend on the missed requests in the rest of the process.
1.c) Another interesting eviction policy is k-LRU, in which $k-1$ virtual LRU caches are put in front of a real one, acting as filters for unpopular content. For renewal traffic, it has been shown that the "Che approximation" works again in this setting and the performance is close that of LFU for large k [46]. Much of the independence structure we have exploited in this work is lost for k-LRU, and again the missed requests in the rest of the process is a relevant object. As a consequence, a rigorous analysis of the hit probability for k-LRU is in our opinion very challenging.
1.d) Finally, we can envisage adding a new layer to the model to represent the chunk request processes. This is relevant since the chunk request present phenomena like "skipping" and interruptions as evidenced by the fact that not all chunk are not equally popular [45]. In consequence, working at the chunk layer may shed light on the impact of the latter phenomena on the LRU cache performance. However, since the chunks are usually requested in order, they have a highly correlated request times, and thus a Poisson model does not seems promising. Thus, the first challenge to address is to propose a pertinent

and tractable model for the chunk request sequences.

2) Parameter Estimation

Our methodology to estimate the popularity distribution of the IRM-M model uses a generic optimization solver. In the non-censored case, Lindsay [43] has shown that the likelihood is a convex function and thus the optimization problem is well-posed. Furthermore, specialized algorithms have been proposed, many of them akin to an Expectation Maximization scheme. Thus, it would be interesting to study the structure of the likelihood in the censored case and adapt the latter algorithms to this setting.

Another possible venue of research is to propose a Maximum Likelihood approach to estimate the distribution of the popularity-lifespan pair for the "Box Model". The estimations in Chapter 2, while effective, are cumbersome and not available for large portion of the dataset. This is due the fact that the lifespan and popularity are hidden variables of the model (as the popularity is hidden for IRM-M). The likelihood in this case is more complex, and more sophisticated methods, such as Monte Carlo approaches [49], may be invoke. Nonetheless, we think that such an approach must be linked to our method for IRM-M since the number of documents in the "Box Model" is also a mixed Poisson random variable.

Chapter 6

Appendices

6.1 Analysis of Simpler Models

The mathematical tools developed in Chapter 3 allow us derive formulas for the transient hit probability for both the IRM and IRM-M models. Additionally we rigorously justify the Che approximation in the case of IRM-M.

IRM Model

For comprehension purposes, we first review in detail the Che approximation method for the stationary hit probability estimation in the IRM model. Given popularities $r_1, r_2, \ldots, r_K$, let $X^k(t)$ denote the number of different documents, apart from the k-th, requested in a time window $[0, t]$, that is,

$$X^k(t) = \sum_{i=1; i \neq k}^{K} \mathbb{1}\{N_i[0,t] \geq 1\}.$$

Let

$$T_C^k = \inf\{t > 0 : X^k(t) \geq C\}$$

be the exit time to level C for process X^k; T_C^k represents the eviction time for content k in a LRU cache of size C, given that it is not requested during this time period. Now, the core of the Che approximation in the stationary case consists in the following steps:

Che.1 assuming that all T_C^k have the same distribution, that is, for each k we have $T_C^k \stackrel{\mathrm{d}}{=} T_C$ for some random time T_C;

Che.2 the random variable T_C is well approximated by a constant t_C called the characteristic time.

The time t_C is implicitly defined by the equation

$$\sum_{k=1}^{K} \mathbb{E}[\mathbb{1}\{N_k[0, t_C] \geq 1\}] = \sum_{k=1}^{K} 1 - e^{-r_k t_C} = C. \tag{6.1}$$

Intuitively, t_C is the time when, on average, C different objects have been requested.

Che.3 The hit probability q_C can then be derived as follows. Using the *Poisson Arrivals See Time Average* property, the hit probability of document k for a cache of size C is equal to $1 - e^{r_k t_C}$, and by averaging on all documents, it follows that

$$q_C \approx \frac{1}{\Lambda} \sum_{k=1}^{K} r_k (1 - e^{-r_k t_C}) \tag{6.2}$$

where $\Lambda = \sum_{k=1}^{K} r_k$.

As for the transient case, we simply assume that $T_C^k \leq W$ as the hit probability does not increase with T_C^k when $T_C^k > W$. Note that we can see the k^{th} request process as one of the Box model with fixed popularity r_k and lifespan W. In consequence, by formula (3.16) the average number of hits for the k^{th} document can be written as

$$\mathbb{E}\left[H_C^k\right] = \mathbb{E}\left[h(r_k, T_C^k)\right]$$

where

$$h(r, t) = (rW - 1)(1 - e^{-rt}) + rte^{-rt}, \quad t < W. \tag{6.3}$$

Thus the transient hit probability $q_C(W)$ is given by

$$q_C(W) = \sum_{k=1}^{K} \mathbb{E}\left[h(r_k, T_C^k)\right].$$

Applying the Che approximation, we then obtain

$$q_C(W) \approx \frac{1}{\Lambda} \sum_{k=1}^{K} r_k (1 - e^{-r_k t_C}) + \frac{1}{\Lambda W} \left(\sum_{k=1}^{K} r_k t_C e^{-r_k t_C} - C \right). \tag{6.4}$$

The second term of (6.4) vanishes as $W \to \infty$, leading to equality (6.2) for the stationary hit probability.

IRM-M Model

We now address the IRM-M case. We first show how to derive the hit probability in this setting; we further prove formally the validity of the Che approximation in the case where $C = \delta K$ and K tends to infinity.

• Given the popularities $R_1, R_2, \ldots, R_K$, let us define X^k, T_C^k as in the previous section, and let $\delta = C/K$ be the proportion of stored documents. As the popularities are here an i.i.d. sample, and since X^k and T_C^k are independent of R_k, the previous quantities do not consequently depend on the document index k. In consequence, this validates the first step of the Che approximation.

For the second step, define the characteristic time t_δ as

$$t_\delta = \varphi^{-1}(\delta) \quad \text{with} \quad \varphi(t) = \mathbb{E}\left[1 - e^{-Rt}\right], \tag{6.5}$$

which is equivalent to dividing both sides of (6.1) by K. Following the same steps as in the previous section, it is easy to derive the following hit probability formulas:

$$q_C \approx \frac{\mathbb{E}\left[R(1 - e^{-Rt_\delta})\right]}{\mathbb{E}[R]}, \tag{6.6}$$

$$q_C(W) \approx \frac{\mathbb{E}\left[R(1 - e^{-Rt_\delta})\right]}{\mathbb{E}[R]} + \frac{\mathbb{E}\left[Rt_\delta e^{-Rt_\delta}\right] - \delta}{\mathbb{E}[R]\, W}. \tag{6.7}$$

Equations (6.6) and (6.7) are the respective IRM-M equivalents to Equations (6.2) and (6.4).

• We now show that the second step of the Che approximation is asymptotically exact, that is, the random variable T_C can be replaced by the associated characteristic time t_δ. Consider the case where the cache size scales with the catalog size, that is, δ remains constant, and C and K grow to infinity. Recall that the distribution of T_C is given by

$$\mathbb{P}[T_C > t] = \mathbb{P}\left[\sum_{k=1}^{K} \mathbb{1}\{N_k[0,t] \geq 1\} < C\right]$$

for $t \geq 0$, which can be rewritten as

$$\mathbb{P}[T_{\delta K} > t] = \mathbb{P}\left[\frac{1}{K}\sum_{k=1}^{K} \mathbb{1}\{N_k[0,t] \geq 1\} < \delta\right]. \tag{6.8}$$

An application of the law of large numbers shows that

$$\lim_{K\to\infty} \frac{1}{K}\sum_{k=1}^{K} \mathbb{1}\{N_k[0,t] \geq 1\} = \varphi(t)$$

almost surely; using (6.8), the exit time $T_{\delta K}$ thus converges in probability to the constant t_δ, for $\delta \in [0, \varphi(W)]$, where $\varphi(W) = \mathbb{E}[K_0]/K$. Note that we can regard the request process as the one of the Box model with popularity R and fixed lifespan W. As in the case of IRM, the formula (3.16) allow us to show expected number of hits $H_C = H_{\delta K}$ satisfies the identity

$$\mathbb{E}[H_{\delta K}] = \mathbb{E}[h(R, T_{\delta K})]\,;$$

with h defined in (6.3). Finally, applying the bounded convergence theorem [62, Sec. 13.6] to the latter identity and dividing by the expected number of requests $\mathbb{E}[R]$ leads to formulas (6.6) and (6.7), as claimed.

6.2 Algorithms

We briefly review the key algorithms we have implemented for this work.

LRU Cache Simulation

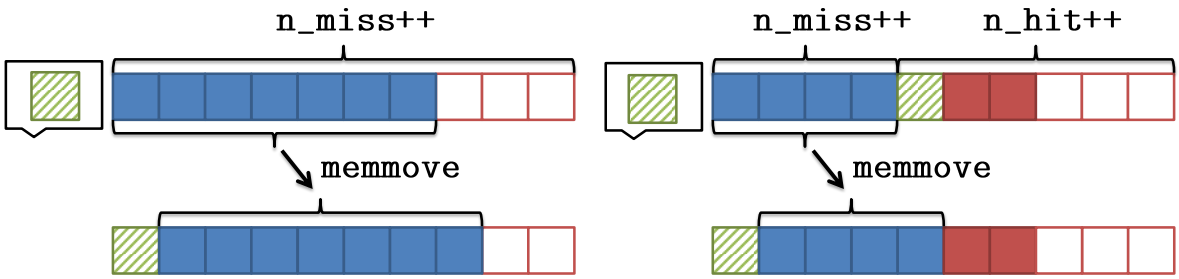
The LRU policy allows the simulation of a cache to be carried out simultaneously for all cache sizes $0 \leq C \leq K_0$, where K_0 is the number of objects in the trace. We achieve this by using a Move-to-Front (MTF) list of size K_0 and observing that, by event equality (1.1), a truncated MTF list at position C behaves exactly like a LRU cache of that size.

We implemented this algorithm in the C Language. First, we allocate an array of size K_0 and initialize it to 0. Upon a request of the document $1 \leq k \leq K_0$, we perform a linear search on the array for it. Then:

- if it is a first document request (Figure 6.1a), we add it to the top of the list while shifting the rest by one slot. We count one miss for document k for all cache sizes;
- if it is already in the array (Figure 6.1b), then we save the location i where it was stored. We put the element in the first slot while shifting the necessary documents by one slot. We count a hit for document k at cache sizes greater than i and one miss in the other cases.

The shifting of objects in the array is implemented by means of the function `memmove`.

We implemented two performance optimizations by keeping track of the documents already in the list and calculating the hits for only a fraction of the possible cache sizes.



(a) *Handling of the first request* (b) *Handling of subsequent requests*

Figure 6.1: *Implementation of the LRU cache simulator*

Mixing Document Request Sequences

A central task for the trace randomizations and simulation of the traffic models is the superposition of the individual document request sequences. For example, to simulate the Box model, it is easy to obtain the non-shifted request times, given a document popularity-lifespan pair and the catalog arrival times. However, to obtain the request trace we must superpose a large number of request sequences.

To perform the superposition efficiently, we implemented a data structure which is a modified *min-priority queue* [14, Sec. 6.5] that allows a quick retrieval of the next request. The original min-priority queue can be regarded as a binary tree where the priority of any node is smaller than of its children. This ensures that the element with smallest value is always at the top of the tree, thus allowing to implement an efficient EXTRACT-MIN method.

Our modification to this data structure consists in replacing single values by the lists of ordered request times (see Figure 6.2). The priority of each list is the time at the top of the list. Thus, upon a call to the EXTRACT-MIN procedure of the queue, it returns the list that has the next request to put in the trace. Then, we extract the first element of this list and, if it is still non-empty, we reinsert it into the queue.

To obtain the trace, we fill the queue with all the individual document request sequences and then empty it by successively calling EXTRACT-MIN. We implemented this strategy in C by modifying the array based implementation in [58, Ch. 8].

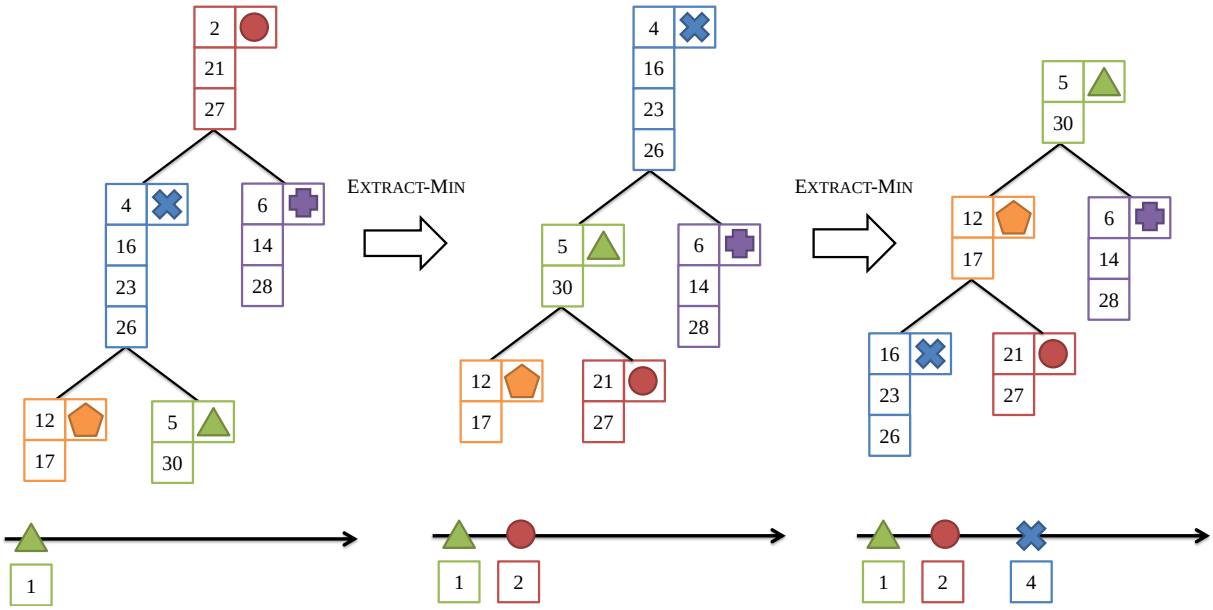


Figure 6.2: Evolution of the modified priority queue and trace after two EXTRACT-MIN calls

Bibliography

[4] B. Ahlgren, C. Dannewitz, C. Imbrenda, D. Kutscher, and B. Ohlman. A Survey of Information-Centric Networking. *Communications Magazine, IEEE*, 2012.

[5] F. Baccelli and P. Brémaud. *Elements of queueing theory: Palm Martingale calculus and stochastic recurrences*. Springer Science & Business Media, 2013.

[6] F. Baccelli, B. Kauffmann, and D. Veitch. Towards Multihop Available Bandwidth Estimation. *ACM SIGMETRICS Performance Evaluation Review*, 2009.

[7] D. S. Berger, P. Gland, S. Singla, and F. Ciucu. Exact analysis of TTL cache networks. *Performance Evaluation*, 2014.

[8] R. Buyya, M. Pathan, and A. Vakali. *Content Delivery Networks*. Springer Science & Business Media, 2008.

[9] Y. Carlinet, T. D. Huynh, B. Kauffmann, F. Mathieu, L. Noirie, and S. Tixeuil. Four Months in DailyMotion: Dissecting User Video Requests. In *8th International Wireless Communications and Mobile Computing Conference (IWCMC)*. IEEE, 2012.

[10] H. Che, Y. Tung, and Z. Wang. Hierarchical Web Caching Systems: Modeling, Design and Experimental Results. *IEEE Journal on Selected Areas in Communications*, 2002.

[11] Cisco Systems, Inc. Cisco Visual Networking Index: Forecast and Methodology, 2014–2019, 2015.

[12] A. Clauset, C. R. Shalizi, and M. E. Newman. Power-law distributions in empirical data. *SIAM Review*, 2009.

[13] F. Comte and V. Genon-Catalot. Adaptive Laguerre density estimation for mixed Poisson models. *Electronic Journal of Statistics*, 2015.

[14] T. H. Cormen, C. E. Leiserson, R. L. Rivest, and C. Stein. *Introduction to Algorithms*. MIT press, 3rd edition, 2009.

[15] D. J. Daley and D. Vere-Jones. *An Introduction to the Theory of Point Processes*, volume 1. Springer, 2nd edition, 2003.

[16] D. J. Daley and D. Vere-Jones. *An Introduction to the Theory of Point Processes*, volume 2. Springer, 2nd edition, 2008.

[17] A. Dembo and O. Zeitouni. *Large Deviations Techniques and Applications*. Springer Science & Business Media, 2009.

[18] P. J. Denning. The Locality Principle. *Communications of the ACM*, 2005.

[19] J. A. dos Santos Gromicho. *Quasiconvex optimization and location theory*. Springer Science & Business Media, 2013.

[20] A. Erramilli, O. Narayan, and W. Willinger. Experimental Queueing Analysis with Long-Range Dependent Packet Traffic. *IEEE/ACM Transactions on Networking*, 1996.

[21] J. A. Fill and L. Holst. On the Distribution of Search Cost for the Move-to-Front Rule. *Random Structures & Algorithms*, 1996.

[22] P. Flajolet, D. Gardy, and L. Thimonier. Birthday paradox, coupon collectors, caching algorithms and self-organizing search. *Discrete Applied Mathematics*, 1992.

[23] N. C. Fofack, P. Nain, G. Neglia, and D. Towsley. Analysis of TTL-based Cache Networks. In *6th International Conference on Performance Evaluation Methodologies and Tools (VALUETOOLS)*. IEEE, 2012.

[24] N. C. Fofack, D. Towsley, M. Badov, M. Dehghan, and D. L. Goeckel. An approximate analysis of heterogeneous and general cache networks. Technical report, Inria, 2014.

[25] G. B. Folland. *Real Analysis: Modern Techniques and their Applications*. John Wiley & Sons, 2nd edition, 1999.

[26] C. Fricker, P. Robert, and J. Roberts. A Versatile and Accurate Approximation for LRU Cache Performance. In *24th International Teletraffic Congress (ITC)*. ACM, 2012.

[27] C. Fricker, P. Robert, J. Roberts, and N. Sbihi. Impact of traffic mix on caching performance in a content-centric network. In *Conference on Computer Communications*. IEEE, 2012.

[28] M. Gallo, B. Kauffmann, L. Muscariello, A. Simonian, and C. Tanguy. Performance evaluation of the random replacement policy for networks of caches. *Performance Evaluation*, 2014.

[29] M. Garetto, E. Leonardi, and S. Traverso. Efficient analysis of caching strategies under dynamic content popularity. In *IEEE Conference on Computer Communications (INFOCOM)*, April 2015.

[30] F. Guillemin, B. Kauffmann, S. Moteau, and A. Simonian. Experimental analysis of caching efficiency for YouTube traffic in an ISP network. In *25th International Teletraffic Congress (ITC)*. IEEE, 2013.

[31] L. Guo, E. Tan, S. Chen, Z. Xiao, and X. Zhang. The stretched exponential distribution of internet media access patterns. In *27th Symposium on Principles of distributed computing*. ACM, 2008.

[32] A. Gut. *Probability: A Graduate Course*. Springer Science & Business Media, 2006.

[33] N. Hohn, D. Veitch, and P. Abry. Cluster Processes: A Natural Language for Network Traffic. *IEEE Transactions on Signal Processing, Special Issue "Signal Processing in Networking"*, August 2003.

[34] C. Imbrenda, L. Muscariello, and D. Rossi. Analyzing Cacheable Traffic in ISP Access Networks for Micro CDN Applications via Content-centric Networking. In *1st International Conference on Information-Centric Networking*, ICN '14. ACM, 2014.

[35] P. R. Jelenković. Asymptotic approximation of the move-to-front search cost distribution and least-recently used caching fault probabilities. *Annals of Applied Probability*, 1999.

[36] P. R. Jelenković and X. Kang. Characterizing the miss sequence of the LRU cache. *ACM SIGMETRICS Performance Evaluation Review*, 2008.

[37] P. R. Jelenković and A. Radovanović. Least-recently-used caching with dependent requests. *Theoretical computer science*, 2004.

[38] S. Jin and A. Bestavros. Sources and characteristics of web temporal locality. In *8th International Symposium on Modeling, Analysis and Simulation of Computer and Telecommunication Systems (MASCOTS)*. IEEE, 2000.

[39] J. Jung, E. Sit, H. Balakrishnan, and R. Morris. DNS Performance and the Effectiveness of Caching. *IEEE/ACM Transactions on Networking*, 2002.

[40] D. Karlis. A General EM Approach for Maximum Likelihood Estimation in Mixed Poisson Regression Models. *Statistical Modelling*, 2001.

[41] J. F. Kurose and K. W. Ross. *Computer Networking: A Top Down Approach*. Pearson, 6th edition, 2013.

[42] E. Leonardi and G. L. Torrisi. Least Recently Used caches under the Shot Noise Model. In *IEEE INFOCOM 2015*, 2015.

[43] B. G. Lindsay. Mixture Models: Theory, Geometry, and Applications. *Institute for Mathematical Statistics: Hayward, CA*, 1995.

[44] P. Loiseau, P. Gonçalves, S. Girard, F. Forbes, and P. Vicat-Blanc Primet. Maximum Likelihood Estimation of the Flow Size Distribution Tail Index from Sampled Packet Data. In *Performance Evaluation Review*. ACM SIGMETRICS, 2009.

[45] L. Maggi, L. Gkatzikis, G. Paschos, and J. Leguay. Adapting caching to audience retention rate: Which video chunk to store? *arXiv preprint arXiv:1512.03274*, 2015.

[46] V. Martina, M. Garetto, and E. Leonardi. A unified approach to the performance analysis of caching systems. In *Proceedings of IEEE INFOCOM*, 2014.

[47] Mathematical Association of America. Problems and Solutions. *The American Mathematical Monthly*, 118(3):pp. 275–282, 2011.

[48] P. D. Miller. *Applied Asymptotic Analysis*. American Mathematical Soc., 2006.

[49] J. Moller and R. P. Waagepetersen. *Statistical Inference and Simulation for Spatial Point Processes*. CRC Press, 2003.

[50] B. N. Oreshkin, N. Regnard, and P. L'Ecuyer. Rate-based daily arrival process models with application to call centers. Technical report, Université de Montréal, 2014.

[51] A. Panagakis, A. Vaios, and I. Stavrakakis. Approximate analysis of LRU in the case of short term correlations. *Computer Networks*, 2008.

[52] D. A. Patterson and J. L. Hennessy. *Computer Organization and Design: The Hardware/Software Interface*. Newnes, 5th edition, 2014.

[53] K. Psounis, A. Zhu, B. Prabhakar, and R. Motwani. Modeling correlations in web traces and implications for designing replacement policies. *Computer Networks*, 2004.

[54] J. Ridoux, A. Nucci, and D. Veitch. Characterization of Wireless Traffic Based on Semi-Experiments. Technical report, LIP6, Université Pierre et Marie Curie, 2005.

[55] J. Roberts and N. Sbihi. Exploring the memory-bandwidth tradeoff in an information-centric network. In *25th International Teletraffic Congress (ITC)*. IEEE, 2013.

[56] G. Rossini and D. Rossi. Coupling caching and forwarding: Benefits, analysis, and implementation. In *1st international conference on Information-centric networking*. ACM, 2014.

[57] F. Roueff and T. Rydn. Nonparametric estimation of mixing densities for discrete distributions. *The Annals of Statistics*, 2005.

[58] R. Sedgewick. *Algorithms in C – Parts 1-4*. Addison-Wesley, third edition, 1998.

[59] S. Traverso, M. Ahmed, M. Garetto, P. Giaccone, E. Leonardi, and S. Niccolini. Temporal locality in today's content caching: Why it matters and how to model it. *Computer Communication Review*, 2013.

[60] S. Vanichpun and A. M. Makowski. Comparing strength of locality of reference – Popularity, majorization, and some folk theorems. In *23rd Annual Joint Conference of the IEEE Computer and Communications Societies (INFOCOM)*. IEEE, 2004.

[61] W. Whitt. *Stochastic-process limits: an introduction to stochastic-process limits and their application to queues*. Springer Science & Business Media, 2002.

[62] D. Williams. *Probability with Martingales*. Cambridge University Press, 1991.

[63] H. Yin, X. Liu, G. Min, and C. Lin. Content delivery networks: a bridge between emerging applications and future IP networks. *Network*, 2010.

université PARIS-SACLAY | École doctorale de mathématiques Hadamard (EDMH)

ÉCOLE POLYTECHNIQUE UNIVERSITÉ PARIS-SACLAY

Titre : Une méthode “data-driven” pour la modélisation de la performance d’un serveur cache
Mots clés : Evaluation de performance, Probabilités, Statistique, Réseaux, Cache

Résumé : La nécessité de distribuer des quantités massives de contenus multi-média à un nombre croissant d’utilisateurs s’est accrue au cours de la dernière décennie. La solution actuelle pour cette demande en croissance constante est fourni par les systèmes connues sous le nom de *Content Delivery Networks*, qui gèrent actuellement la majorité du trafic multi-média en utilisant une architecture distribuée. Ce problème de distribution a également motivé l’étude de nouvelles solutions tel que celui proposé par l’*Information Centric Networking*, dont l’objectif est d’ajouter des capacités de livraison de contenus à la couche réseau, moyennant un découplage des données et de leur localisation. Dans ces deux architectures, les serveurs cache jouent un rôle clé, en permettant un usage efficace des ressources de réseau pour la distribution de contenus. En conséquence, l’étude des techniques pour l’évaluation des performances des serveurs cache a trouvé un nouvel élan ces dernières années.

Dans cette thèse, nous proposons un cadre complet pour la modélisation des performances d’un cache utilisant la politique de remplacement *Least Recently Used* (LRU). Notre cadre considère, outre l’analyse mathématique, deux procédures qui relient les données au modèle : Dans la première procédure, nous proposons un modèle simple qui est a priori représentatif des caractéristiques essentielles du trafic mesuré; dans la deuxième nous estimons les paramètres du modèle à partir des traces de trafic. Les contributions de cette thèse concernent chacune des procédures mentionnées.

Title: A Data-Driven Approach for Cache Performance Modelling
Keywords: Performance Evaluation, Probability, Statistics, Networks, Caching

Abstract: The need to distribute massive quantities of multimedia content to multiple users has increased tremendously in the last decade. The current solution to this ever-growing demand are *Content Delivery Networks*, that handle nowadays the majority of multimedia traffic by means of a distributed architecture. This distribution problem has also motivated the study of new solutions such as the *Information Centric Networking* paradigm, whose aim is to add content delivery capabilities to the network layer by decoupling data from its location. In both architectures cache servers play a key role, allowing efficient use of network resources for content delivery. As a consequence, the study of cache performance evaluation techniques has found a new momentum in recent years.

In this dissertation, we propose a framework for the performance modeling of a cache ruled by the *Least Recently Used* (LRU) discipline. Our framework is data-driven in the sense that, in addition to the usual mathematical analysis, we address two additional data-related problems: the first one is to propose a model that is a priori both simple and representative of the essential features of the measured traffic. The second one is the estimation of the model parameters starting from traffic traces. The contributions of this thesis concerns each of the above tasks.

Université Paris-Saclay
Espace Technologique / Immeuble Discovery
Route de l’Orme aux Merisiers RD 128 / 91190 Saint-Aubin, France