A LYAPUNOV ANALYSIS OF LRU

BY

MICHAEL BRENNER

THESIS

Submitted in partial fulfillment of the requirements
for the degree of Master of Science in Electrical and Computer Engineering
in the Graduate College of the
University of Illinois at Urbana-Champaign, 2020

Urbana, Illinois

Adviser:

Professor Rayadurgam Srikant

ABSTRACT

Caches are segments of memory that store requested information in a system subject to a set of decision rules, defined as the caching algorithm. One of the most popular caching algorithms is the least recently used algorithm (LRU) due to its simplicity and effectiveness in a multitude of applications. LRU caches operate by storing objects in the order that they were most recently requested. Further, whenever an item is requested that is not currently in the cache, the requested item is placed at the head of the cache, and the least recently requested item is evicted. Many have suggested a tie between the performance of an LRU cache and a time to live (TTL) cache. In this thesis, we present a unique Lyapunov based proof for an asymptotically exact TTL approximation for the steady state distribution of our LRU Markov model. We further present ongoing theoretical extensions to other variants of LRU, as well as simulations that validate our model. We conclude by proposing a variance corrected model to better approximate hit rate over time.

To my parents, for their love and support.

ACKNOWLEDGMENTS

I would like to take a moment to thank everyone who helped with this thesis. First, I would like to thank my advisor, Professor Srikant, for helping me choose this problem and providing guidance throughout my research. I would also like to highlight that this project was performed in collaboration with Siddhartha Satpathi who is currently a PhD student advised by Professor Srikant. Siddhartha directed me through the proofs presented in this thesis and provided significant experience and advice during my research.

TABLE OF CONTENTS

CHAPTER 1

INTRODUCTION

Caching systems are used to increase performance over a network by storing highly requested items at a local position. Specifically, a cache is a segment of local memory that is dedicated to maintain frequently requested information. For example, consider a social network. Suppose an item is requested multiple times by different users. A naive approach for storage in this problem would be to service each request uniquely. Thus, after each request, the item must be located in a server and supplied to the user. This approach is highly inefficient since multiple searches are performed for the same item. As a solution, network designers include caching schemes to minimize the number of searches for a given item. Once an item is initially requested, the item is copied to the cache to allow for similar requests to be serviced locally.

A key design feature is determining what items are to be included in the cache. One of the classic algorithms that serves as a foundation for many systems is the least recently used (LRU) algorithm. This algorithm can be thought of as follows. Consider a cache that can hold m unique items. The items are positioned in the order that the requests arrive, implying that the first position is the most recently requested object while the m^{th} object is the oldest. Assume that at the current time, the cache has all m object positions occupied. Now, suppose a new request arrives to the cache. If the requested item is currently in the cache, say at position $i \in \{1, 2..., m\}$, the request is serviced and the item is moved from the i^{th} position to the first position. All items that were previously in positions 1 through $i - 1$ are shifted back one position. Now, suppose that another request arrives for an item that is not in the cache. The system will initially check if the item is in the cache and deduce that it is not. The requested object is then found in the general storage and serviced. Additionally, the object is copied to the first position of the cache and all objects that were previously in positions 1 through $m-1$ are shifted one position. The m^{th} item is evicted. This caching algorithm

has proven quite effective and is frequently used across many domains.

In general, the metric used to evaluate different caching algorithms is known as the hit rate, $h(t)$. This quantity represents the probability that a requested item currently resides in the cache at time t. A high hit rate is desirable as it implies that the cache is effective at discerning between popular and unpopular items. We model this system as a Markov chain and we wish to quantify the steady state behavior of the LRU caching algorithm by demonstrating convergence of the hit rate.

Fagin [1] first proposed a surprising characterization of the LRU algorithm that was rediscovered by Che et al. [2] nearly 25 years later. The main focus of these works was demonstrating that objects in LRU caches perform asymptotically similar to that of a time to live, or TTL, cache given independent references. Essentially, this model exemplifies that given a new item is requested at time t, a timer is set with characteristic time T_c. If the same item is requested before time $t + T_c$, then the item remains in the cache and the timer is reset. If no request comes, then the item is evicted when the timer reaches T_c. Thus the probability that a requested item is currently in the cache can be modeled as an exponential random variable with a rate that is proportional to T_c and the item's request probability. This result came to be known as the Fagin-Che approximation. Under a similar model, our goal is to prove that this TTL approximation is an accurate representation of the LRU caching algorithm.

We then wish to show how partitioning the cache can lead to significant performance increase. LRU(m), or segmented LRU, was another algorithm introduced to improve LRU by taking both time and frequency into account when caching. The fundamental performance of this algorithm can be described as m unique LRU caches chained together. Given an item is requested, if the item is not currently in the cache, then it is inserted at the head of list 1. Otherwise, if the item is currently in subcache $i \in \{1, 2, \ldots, m\}$, the item is promoted to the head of list $\max(m, i+1)$. Thus, items that are more frequently requested are promoted to higher lists than other items. By incorporating this frequency into the caching scheme, the hit rate performance of the cache is significantly improved. We speculate on possible shortcomings of existing TTL approximations for LRU(m).

1.1 Contributions

The contributions of this thesis can be summarized in distinct sections. We initially present a unique proof for the convergence of LRU. We then extend our model to segmented LRU and speculate on possible improvements. These contributions are summarized below:

1. We first create a Markov model that captures the dynamics of any caching system. From this model, we then demonstrate convergence of the LRU system to the TTL approximation using classic Lyapunov theory and connections to the classic coupon collector problem. We conclude our analysis of LRU by simulating our model and demonstrating convergence to the TTL approximation.

2. We introduce a separate Markov model for easy generalization to more complex caching algorithms. Namely, we present an alternative characterization of LRU that is directly related to LRU(m). We then present the theoretical solution to this model, and explain its significance.

3. We provide simulation results that demonstrate the accuracy of our model under the hit rate metric.

4. We speculate on adding variance correction terms to our TTL approximation to better capture the current dynamics of our system. We leave this as a continued problem.

1.2 Related Work

The least recently used algorithm is a fundamental caching algorithm which is frequently used due to its simplicity and effectiveness. Further, its seemingly simple dynamics make it a common algorithm to analyze using Markovian analysis. Fagin [1] first analyzed caching systems and developed a connection between TTL and LRU caches. Che et al. [2] rediscovered this result much later as caching systems gained prominence with the rise of the internet. Since Che et al. and Fagin introduced the approximation, there has been a lot of interest in LRU's steady state performance. Recall that the Fagin-Che approximation demonstrates that the performance of LRU caches

can be approximated using a simple TTL cache. Fricker et al. first analyzed the Fagin-Che approximation in [3]. This paper associated the independent reference model (IRM) to LRU, which has been a cornerstone of subsequent analysis. This model asserts that item requests are made as an infinite sequence of independent random variables. Requests are made subject to an underlying probability distribution over all items. Analysis was performed by characterizing the request time of each item as an exponential random variable with rate proportional to the probability that the item would be selected. Fricker et al. ultimately demonstrated convergence results through central limit theorem arguments that proved the concentration of arrival times for objects in the cache.

After the initial implementation of LRU, many more variants were introduced that demonstrated significant performance improvement. One particular variant, segmented LRU (also called LRU(m)), has recently received increasing attention due to its performance benefits. Segmented LRU was proposed by Karedla et al. in [4] as a means of rewarding items that were requested more than once in an LRU scheme. Karedla et al. proposed that by partitioning the cache once, hit rate improved at no additional cost. Since the publication of this paper, segmented LRU has remained prominent with increased partitioning. For example, Facebook currently employs a variant of segmented LRU with three partitions to cache user data. By switching to a segmented LRU algorithm, they observed a 15.5% increase in hit rate compared to the existing algorithm [5].

There has been much work to expand the TTL approximation that held for LRU to more complex settings. Many researchers applied Markovian analysis to develop concentration results for this family of algorithms. Gast and Van Houdt developed a particular Markov model in [6] that motivated this thesis. In the work, Gast and Van Houdt demonstrated that the steady state distribution for LRU could be approximated by the fixed point of a collection of ordinary differential equations. This fixed point was precisely the TTL approximation. This result was generalized to segmented LRU, as a fixed point solution was proven such that each subcache had a distinct TTL approximation.

1.3 Thesis Overview

In this work, we begin by presenting a model similar to that of Gast and Van Houdt and summarize our main results in Chapter 2. We then prove these results in Chapter 3 through a Lyapunov-based proof for the convergence of LRU performance to the TTL approximation. In Chapter 4, we expand our model to LRU(m), before simulating the model in Chapter 5 and speculating about future work in Chapter 6.

CHAPTER 2

LRU MODEL AND MAIN RESULT

2.1 LRU Model

The main objective of this section is to introduce the Markov model used throughout this section. This model was motivated by Gast and Van Houdt in [6]. Consider a population with two types of objects, referred to as Type 1 objects and Type 2 objects. Suppose that the total size of the population is precisely $2N$, such that there are N distinct objects of each type. These objects are to be efficiently cached in an LRU caching system. Objects of each type arrive to the cache at a rate α_i with $i = 1, 2$, such that $\alpha_1 + \alpha_2 = 1$. Thus, each distinct object k will have request popularity p_k defined as:

$$p_k = \begin{cases} \frac{\alpha_1}{N} \text{ if object k is Type 1} \\ \frac{\alpha_2}{N} \text{ if object k is Type 2} \end{cases} \tag{2.1}$$

In order to accurately capture scaling dynamics similar to those of a true system, assume that a new request will arrive at the cache every $\frac{1}{N}$ time units. The intuition of this scaling is that as more objects enter the system and the system becomes more complex, there will be more requests. We will assume throughout this thesis that all requests are made independently, and that all items are of equal size. Let us further define the Markov state vector, $y_{i,b}(t)$, as the fraction of type i items that have been requested in the past b time units from time t for all b.

Using these properties, we can define the Markov chain of this system as follows:

$\big(y_{i,b}(t+\frac{1}{N}), y_{2,b}(t+\frac{1}{N})\big) =$

$$\begin{cases} \big(y_{1,b}(t)+\frac{1}{N}-\tilde{y}_{1,b}(t)\big), \big(y_{2,b}(t)-\tilde{y}_{2,b}(t)\big) & \text{w.p.} \quad \alpha_1(1-y_{1,b}(t)) \\ \big(y_{1,b}(t)-\tilde{y}_{1,b}(t)\big), \big(y_{2,b}(t)+\frac{1}{N}-\tilde{y}_{2,b}(t)\big) & \text{w.p.} \quad \alpha_2(1-y_{2,b}(t)) \\ \big(y_{1,b}(t)-\tilde{y}_{1,b}(t)\big), \big(y_{2,b}(t)-\tilde{y}_{2,b}(t)\big) & \text{w.p.} \quad \alpha_1 y_{1,b}(t)+\alpha_2 y_{2,b}(t) \end{cases} \tag{2.2}$$

where we define

$$\tilde{y}_{i,b}(t) = y_{i,b}(t) - y_{i,b-\frac{1}{N}}(t)$$

This expression represents a history term to describe the m^{th} item in the cache. Observe that $\tilde{y}_{i,b}(t)$ will equal $\frac{1}{N}$ if the m^{th} item is type i and 0 else. Using this definition, the provided Markov chain can be broken down into three intuitive situations. The first two cases correspond to the event that an object is requested that is not currently in the cache. The third event corresponds to a hit, or the event that the object requested currently resides in the cache. This Markov chain will serve as the foundation for all subsequent analysis.

Further, note that the boundary conditions for the above Markov chain are given by

$$y_{i,b}(t) = 0, \quad \forall b \geq t$$

Given this Markovian model, we now wish to characterize the steady state behavior of the system. Formally, we consider the drift of the Markov chain in steady state:

$$\mathbb{E}\Big[y_{i,b}\big(t+\frac{1}{N}\big) - y_{i,b}(t)\Big] = 0$$

Given the entire history of the caching system up to time t, denoted as $y(t)$, the drift can be characterized through the following using nested expectations:

$$\mathbb{E}\Big[\mathbb{E}\Big[y_{i,b}\big(t+\frac{1}{N}\big) - y_{i,b}(t)\Big|y(t)\Big]\Big] = \mathbb{E}\Big[-\tilde{y}_{i,b}(t) + \frac{1}{N}\alpha_i(1-y_{i,b}(t))\Big]$$

We wish to show that this conditional expectation converges to the fixed point of a partial differential equation. To obtain this approximate PDE, consider dividing the above equation by $\frac{1}{N}$ and taking the limit as $\frac{1}{N} \to 0$. The drift term and the $\tilde{y}_{i,b}$ term both now appear similar to a partial derivative, while the remaining term is unaffected by the limit. Thus, we arrive at the

following PDE:

$$\frac{\partial y_{i,b}}{\partial t} = -\frac{\partial y_{i,b}}{\partial b} + \alpha_i\big(1 - y_{i,b}(k)\big)$$

In steady state, the partial with respect to time will be zero and we note that the solution to this ordinary differential equation has an exponential form as [2] suggested. Note that the TTL approximation will be denoted $y^*_{i,b}$ throughout this thesis.

$$y^*_{i,b} = 1 - e^{-\alpha_i b}$$

The remainder of this chapter presents theorems that demonstrate the convergence of the Markov chain to this fixed point solution to the ODE. Note that these states, $y_{i,b}(t)$, are important since they uniquely characterize the hit rate. The connection to hit rate will be presented in Section 2.2.2.

2.2 Main Result

In this section, the main theorems that demonstrate the convergence of the model provided in Section 2.1 to the TTL approximation will be motivated and presented. The order of presentation is as follows:

- First we will derive a Lyapunov function that will help prove the convergence of $y_{i,b}$ to $y^*_{i,b}$ for any fixed b.
- We will then formalize the notion of hit rate and provide a bound on the absolute difference between the predicted hit rate and the approximate hit rate.
- Finally, we will draw a connection to the classic coupon collector problem to complete the proof and demonstrate that the steady state hit rate of the LRU caching algorithm converges to the TTL approximation.

Note that the formal proofs of these theorems will be presented in the following chapter.

2.2.1 Lyapunov Approach

In order to bound the difference between $y_{i,b}$ and $y^*_{i,b}$, a Lyapunov function, $V(y_{i,b})$, must be found such that

$$\mathbb{E}\Big[V\Big(y_{i,b}\big(t+\frac{1}{N}\big)\Big) - V(y_{i,b}(t))\Big] = 0$$

The objective is thus to find a suitable function V that will establish a bound between the model and the TTL approximation. It was determined that the proper Lyapunov function was:

$$V(y_{i,b}) = \frac{1}{2\alpha_i}\big(y_{i,b} - (1 - e^{-\alpha_i b})\big)^2 \tag{2.3}$$

Using this V, the second moment of the difference between the TTL approximation and the true system can be bounded.

We now present a theorem that demonstrates a bound on the concentration of the Markov state variable, $y_{i,b}$.

Theorem 1. *If a system, $y_{i,b}$, described by the Markovian model in* (2.2) *is currently in steady state, then for any fixed b, the following bound holds:*

$$\mathbb{E}[(y_{i,b} - y^*_{i,b})^2] \leq \frac{6}{Na_i},\ for\ i = 1, 2$$

The proof of Theorem 1 follows directly from manipulation of the Lyapunov function defined by (2.3). After grouping all dominant $O(1/N)$ terms and letting all higher order terms go to 0, a telescoping sum can be performed to arrive at the given bound.

This result allows us to deduce that for each fixed b, the Markov chain will converge to a form similar to the TTL approximation. We will now extend this result to characterize the hit rate.

2.2.2 Hit Rate Definition

After demonstrating that the steady state approximation is close to the true dynamics of the system, we can develop an expression for hit rate. Recall that hit rate is defined as the probability that a newly requested item is in the cache. Traditionally, the hit rate is calculated by summing the rate of

arrivals for items in the cache. To collect all items that are currently in a cache of size m, we will define a random variable $\theta(t)$ as:

$$\theta(t) = \inf\Big(b : \sum_{i=1}^{2} y_{i,b} \geq \frac{m}{N}\Big) \tag{2.4}$$

This inequality is essentially computing $\theta(t)$ by starting at time t and moving back the minimum number of time steps until the cache is full and holds m unique objects. Note that by summing $y_{1,b}(t)$ and $y_{2,b}(t)$, we have an expression for the fraction of unique items that have been requested in the previous b steps. Therefore, when this fraction equals $\frac{m}{N}$, the cache is full. Using this result, we can then define the hit rate, $h(t)$, as

$$h(t) = \alpha_1 \cdot y_{1,\theta(t)}(t) + \alpha_2 \cdot y_{2,\theta(t)}(t) \tag{2.5}$$

Previously in Theorem 1, we have shown that $y_{i,b} \to y^*_{i,b}$. Using this fact, we must demonstrate that the above hit rate converges to the approximate hit rate that is defined from the definition of $y^*_{i,b}$:

$$h^* = \alpha_1 \cdot \big(1 - e^{-\alpha_1 \cdot T_m}\big) + \alpha_2 \cdot \big(1 - e^{-\alpha_2 \cdot T_m}\big) \tag{2.6}$$

where T_m is the solution to

$$m = N \sum_{k=1}^{2} \big(1 - e^{-\alpha_i T_m}\big) \tag{2.7}$$

Note that T_m is classically defined as the characteristic time of the cache. Under a TTL setting, this quantity is the duration that the timer will run before the item is evicted. We thus wish to prove that $h(t)$ tends to h^* as $t \to \infty$, demonstrating that the true caching performance is similar to the TTL approximation. This result is given in the following theorem:

Theorem 2. *Let $h(\infty)$ represent the steady state hit rate of the system described in* (2.2)*. Further, let h^* be defined as* (2.6)*. Then the following bound holds:*

$$\mathbb{E}\big[\big|h(\infty) - h^*\big|\big] \leq O\Big(\frac{1}{N^{\frac{1}{12}}}\Big)$$

We have thus demonstrated that as N, the number of objects, grows large, and $t \to \infty$, the true steady state hit rate will converge to the TTL approx-

imation. This result is proven with the aid of two Lemmas which will be stated and proved in Chapter 3.

CHAPTER 3

PROOFS

In this chapter, we will present formal proofs of Theorem 1 and Theorem 2. Further, two lemmas will be introduced that will be useful in proving Theorem 2.

3.1 Theorem 1 Proof

Recall that Theorem 1 establishes the following bound:

$$\mathbb{E}[(y_{i,b} - y^*_{i,b})^2] \leq \frac{6}{Na_i},\ for\ i = 1, 2$$

To begin the proof, we can apply the Lyapunov function that was proposed in (2.3) to the drift expression of the Markov Chain.

$$\mathbb{E}[(y_{i,b}(t + \frac{1}{N}) - y^*_{i,b})^2 - (y_{i,b}(t) - y^*_{i,b})^2] = 0$$

We can now condition this drift expression on $y_{i,b}(t)$ and expand the $y_{i,b}(t+\frac{1}{N})$ term in accordance with the Markov model. After performing this expansion, we get:

$$\begin{aligned}\mathbb{E}\big[2(y_{i,b}(t) - y^*_{i,b})(\tilde{y}_{i,b}(t) - \frac{P_i}{N})\big] &= \mathbb{E}\big[(\tilde{y}_{i,b}(t))^2 + \frac{P_i}{N^2} - 2\tilde{y}_{i,b}(t)\frac{P_i}{N}\big] \\ &\leq \frac{2}{N^2} \qquad (3.1)\end{aligned}$$

since $\tilde{y}_{i,b}(t) \in \{0, \frac{1}{N}\}$. Note that for ease of notation, P_i was used to replace $\alpha_i(1 - y_{i,b}(t))$. To further simplify notation, for the remainder of this proof, the time variable, t, will be left off as the system is assumed to be in steady state.

We will first show that $2(y_{i,b} - y^*_{i,b})(\tilde{y}_{i,b} - \frac{P_i}{N})$ is close to the difference

$(y_{i,b}(k)-y^*_{i,b})^2-(y_{i,b-\frac{1}{N}}(k)-y^*_{i,b-\frac{1}{N}})^2$. We will then use this result to perform a telescoping sum on the difference to obtain the desired bound. Observe:

$$\begin{aligned}
&2e^{2\alpha_i b}(y_{i,b}-y^*_{i,b})(\tilde{y}_{i,b}-\frac{P_i}{N}) = \\
&\quad = 2e^{\alpha_i b}(y_{i,b}-y^*_{i,b})(e^{\alpha_i b}(y_{i,b}-y^*_{i,b})-e^{\alpha_i(b-\frac{1}{N})}(y_{i,b-\frac{1}{N}}-y^*_{i,b-\frac{1}{N}})) \\
&\quad + 2e^{2\alpha_i b}(y_{i,b}-y^*_{i,b})((1-y_{i,b})(1-\frac{\alpha_i}{N}-e^{-\frac{\alpha_i}{N}})+\tilde{y}_{i,b}(1-e^{-\frac{\alpha_i}{N}})) \\
&\quad \geq 2e^{\alpha_i b}(y_{i,b}-y^*_{i,b})(e^{\alpha_i b}(y_{i,b}-y^*_{i,b})-e^{\alpha_i(b-\frac{1}{N})}(y_{i,b-\frac{1}{N}}-y^*_{i,b}-\frac{1}{N}))-4\frac{e^{2\alpha_i b}}{N^2}
\end{aligned} \tag{3.2}$$

where the final inequality follows from the fact that $1-\frac{\alpha_i}{N}-e^{-\frac{\alpha_i}{N}} \geq -\frac{1}{N^2}$. For ease of notation, we will make the following change of variables:

$$\bar{y}_{i,b} = y_{i,b}-y^*_{i,b}$$

$$\bar{y}_{i,b-\frac{1}{N}} = y_{i,b-\frac{1}{N}}-y^*_{i,b}$$

We can continue to partition this expression to arrive at a difference of squares equation:

$$\begin{aligned}
&2e^{\alpha_i b}\bar{y}_{i,b}\left(e^{\alpha_i b}\bar{y}_{i,b}-e^{\alpha_i(b-\frac{1}{N})}\bar{y}_{i,b-\frac{1}{N}}\right) \\
&\quad = \left(e^{\alpha_i b}\bar{y}_{i,b}+e^{\alpha_i(b-\frac{1}{N})}\bar{y}_{i,b-\frac{1}{N}}\right)\cdot\left(e^{\alpha_i b}\bar{y}_{i,b}-e^{\alpha_i(b-\frac{1}{N})}\bar{y}_{i,b-\frac{1}{N}}\right) \\
&\quad + \left(e^{\alpha_i b}\bar{y}_{i,b}-e^{\alpha_i(b-\frac{1}{N})}\bar{y}_{i,b-\frac{1}{N}}\right)^2 \\
&\quad \geq \left(e^{\alpha_i b}(\bar{y}_{i,b})\right)^2-\left(e^{\alpha_i(b-\frac{1}{N})}(\bar{y}_{i,b-\frac{1}{N}})\right)^2
\end{aligned} \tag{3.3}$$

We can now substitute the change of variables back into (3.3) to obtain:

$$\geq \left(e^{\alpha_i b}(y_{i,b}-y^*_{i,b})\right)^2-\left(e^{\alpha_i(b-\frac{1}{N})}\left(y_{i,b-\frac{1}{N}}-y^*_{i,b-\frac{1}{N}}\right)\right)^2$$

Combining (3.2) and (3.3) with (3.1), we have

$$\mathbb{E}[\left(e^{\alpha_i b}(y_{i,b}-y^*_{i,b})\right)^2-\left(e^{\alpha_i(b-\frac{1}{N})}(y_{i,b-\frac{1}{N}}-y^*_{i,b-\frac{1}{N}})\right)^2] \leq \frac{6e^{2\alpha_i b}}{N^2} \tag{3.4}$$

At this point, we are setting up a telescoping sum over the two squared terms in (3.4). To complete this proof, sum both sides with respect to b over the

domain $\{\frac{1}{N}, \frac{2}{N}, \ldots, b\}$.

$$\sum_{b\in\{\frac{1}{N},\ldots,b\}} \mathbb{E}[\left(e^{a_i b}(y_{i,b} - y^*_{i,b})\right)^2 - \left(e^{a_i(b-1/N)}(y_{i,b-1/N} - y^*_{i,b-1/N})\right)^2] \leq \sum_{b\in\{\frac{1}{N},\ldots,b\}} \frac{6e^{2a_i b}}{N^2}$$

Note that the boundary conditions for this problem are given as $y_{i,0} = y^*_{i,0} = 0$. So,

$$\mathbb{E}[(e^{\alpha_i b}(y_{i,b} - y^*_{i,b}))^2] \leq \frac{6e^{2\alpha_i b}}{N\alpha_i} \tag{3.5}$$

We thus arrive at the prescribed bound:

$$\mathbb{E}[\left(y_{i,b} - y^*_{i,b}\right)^2] \leq \frac{6}{N\alpha_i}$$

This proof thus establishes that $y_{i,b} \to y^*_{i,b}$ as $N \to \infty$.

3.2 Theorem 2 Preliminaries

Theorem 2 establishes the convergence of the true steady state hit rate to the hit rate approximation from the TTL. Before proving this result, we will first establish a relationship between the hit rate, $h(t)$ as defined in (2.5), and the proposed TTL hit rate, h^* as defined in (2.6). This result is given in the following lemma.

Lemma 1.

$$|h(t) - h^*| \leq 2|y_{1,\theta(t)}(t) - y^*_{1,\theta(t)}| + 2|y_{2,\theta(t)}(t) - y^*_{2,\theta(t)}|$$

Proof. Before beginning the proof, observe the true meaning of the $y_{i,\theta(t)}$ term. Given the definition of the state variable and the meaning of $\theta(t)$, as presented in (2.4), we can deduce that $y_{i,\theta(t)}$ represents the fraction of type i items that have been requested in the interval $[t - \theta(t), t]$. Further, because $\theta(t)$ is the request time of the oldest object in the cache, $y_{i,\theta(t)}$ is the fraction of type i items located in the cache at time t. We can now write the absolute

difference of the hit rate explicitly:

$$\begin{aligned}|h(t) - h^*| &= \left|\alpha_1(y_{1,\theta(t)} - y^*_{1,T_m}) + \alpha_2(y_{2,\theta(t)} - y^*_{2,T_m})\right| \\ &\leq \alpha_1\left|y_{1,\theta(t)} - y^*_{1,T_m}\right| + \alpha_2\left|y_{2,\theta(t)} - y^*_{2,T_m}\right| \\ &\leq \left|y_{1,\theta(t)} - y^*_{1,T_m}\right| + \left|y_{2,\theta(t)} - y^*_{2,T_m}\right| \end{aligned} \tag{3.6}$$

where the last line follows from the fact that both α_1 and α_2 are less than 1. We now employ triangle inequality and make the following substitution:

$$|y_{i,\theta(t)} - y^*_{i,T_m}| \leq |y_{i,\theta(t)} - y^*_{i,\theta(t)}| + |y^*_{i,\theta(t)} - y^*_{i,T_m}|$$

Making this substitution in (3.6):

$$|h(t) - h^*| \leq \sum_{i=1}^{2} |y_{i,\theta(t)} - y^*_{i,\theta(t)}| + \sum_{i=1}^{2} |y^*_{i,\theta(t)} - y^*_{i,T_m}| \tag{3.7}$$

Recall that $y^*_{i,b}$ is an exponential, monotonically increasing function in b. We thus know $y^*_{i,\theta(t)} - y^*_{i,T_m}$ must have the same sign for $i = 1, 2$. The sum can therefore be brought inside the absolute value in the final term of (3.7):

$$\sum_{i=1}^{2} |y^*_{i,\theta(t)} - y^*_{i,T_m}| = \left|\sum_{i=1}^{2} y^*_{i,\theta(t)} - y^*_{i,T_m}\right| \tag{3.8}$$

Recall that from the definitions of $\theta(t)$ in (2.4) and T_m in (2.7):

$$y_{1,\theta(t)} + y_{2,\theta(t)} = y^*_{1,T_m} + y^*_{2,T_m} = \frac{m}{N}$$

We can use this relationship to bound (3.8):

$$\begin{aligned}\left|\sum_{i=1}^{2} y^*_{i,\theta(t)} - y^*_{i,T_m}\right| &= \left|y^*_{1,\theta(t)} + y^*_{2,\theta(t)} - y^*_{1,T_m} - y^*_{2,T_m}\right| \\ &= \left|y^*_{1,\theta(t)} + y^*_{2,\theta(t)} - y_{1,\theta(t)} - y_{2,\theta(t)}\right| \\ &\leq \left|y^*_{1,\theta(t)} - y_{1,\theta(t)}\right| + \left|y^*_{2,\theta(t)} - y_{2,\theta(t)}\right| \end{aligned} \tag{3.9}$$

Combining (3.7) with (3.9), we arrive at the desired bound.

$$|h(t) - h^*| \leq 2|y_{1,\theta(t)}(t) - y^*_{1,\theta(t)}| + 2|y_{2,\theta(t)}(t) - y^*_{2,\theta(t)}|$$

Note that we can further upper bound $|h(t) - h^*|$ by taking the supremum over all possible b, which will prove useful in subsequent proofs.

$$|h(t) - h^*| \leq 2\sum_{i=1}^{2} |y_{i,\theta(t)} - y^*_{i,\theta(t)}| \leq 2\sup_b \sum_{i=1}^{2} |y_{i,b} - y^*_{i,b}|$$

□

3.3 Connection to Coupon Collector Problem

We now wish to develop a concentration result on the characteristic time. For mathematical convenience, we will make the following generalizations about our caching model. Suppose our cache size is written as $m = 2\beta N$ with $\beta \in [0, 1]$. Recall that $\theta(t)$, as defined in (2.4), represents the amount of time we need to wait to observe m distinct object requests. Define $\theta(\infty)$ to be the steady state distribution of $\theta(t)$. We thus want to demonstrate concentration of $\theta(\infty)$. We can now make a connection to the coupon collector problem. The basic idea of the coupon collector problem is to determine the number of coupons that need to be drawn with replacement in order to observe all possible coupons. For our result, we are concerned with the partial observance coupon collector variant. In this problem, we are interested in the number of coupons that need to be drawn with replacement in order to observe m distinct coupons. Note that each coupon is drawn from a population of size $2N > m$, and is independent of all other selections. The connection between the LRU algorithm and the coupon collector was initially proposed in [7]. We can therefore use the coupon collector problem to gain further insights on our LRU model.

Lemma 2. *Define $\alpha_{max} = \max(\alpha_1, \alpha_2)$. Then for any $\epsilon > 0$, the following concentration result holds for the steady state distribution of the characteristic time, $\theta(\infty)$:*

$$\mathbb{P}\left(\theta(\infty) \in \left[\mathbb{E}[\theta(\infty)] - \frac{1}{N^{0.5+\epsilon}}, \mathbb{E}[\theta(\infty)] + \frac{1}{N^{0.5+\epsilon}}\right]\right) \geq 1 - \frac{2N^{-2\epsilon}}{\alpha_{max}(1 - 2\alpha_{max}\beta)}$$

Proof. As a brief overview of the proof, we will begin by bounding the vari-

ance of $\theta(\infty)$ and then apply Chebyshev's inequality to arrive at the concentration result. We will exploit the fact that each successive arrival is independent in order to decompose $\theta(\infty)$ into the sum of geometric random variables.

We begin by defining the collection of independent random variables, $X_1, X_2, \ldots, X_m$. Each random variable, X_i, corresponds to the number of time slots between the $i-1^{th}$ request and the i^{th} request. For example, X_1 represents the number of time slots needed to see the arrival of the first item. Because the system is guaranteed to have an arrival in each time slot, X_1 will equal 1 with probability 1. X_2 is a random variable that represents the number of time slots required to observe a second unique object that is different from the first. Thus, each X_i is a geometric random variable with the probability of success equal to one minus the sum of request probabilities of all distinct items that have been observed in the time slots associated with $X_1, X_2, \ldots, X_{i-1}$.

The definition of X_i is useful because it allows for the decomposition of $\theta(\infty)$ into the following sum:

$$\theta(\infty) = X_1 + X_2 + \ldots + X_m$$

Define q_i as the request probability of item i that corresponds to X_i. Then each X_i is distributed:

$$X_i \sim Geo(1 - \sum_{j=1}^{i-1} q_j)$$

Since each X_i is independent of all other X_i's, we can express the variance of $\theta(\infty)$ as the sum of the variance of each X_i.

$$Var[\theta(\infty)] = \sum_{i=1}^{m} Var[X_i]$$

Now, in order to obtain a bound on the variance of T, we can use the following weak upper bound as it proves sufficient for our application.

$$\begin{aligned} Var[\theta(\infty)] &= \mathbb{E}[\theta(\infty)^2] - \mathbb{E}[\theta(\infty)]^2 \\ &\leq \mathbb{E}[\theta(\infty)^2] \end{aligned}$$

We therefore must compute the second moment of $\theta(\infty)$, and thus of X_i.

Define $\mathcal{S}_i$ to be the set of all possible unique request sequences of length $i-1$. From the law of total expectation:

$$\mathbb{E}[X_i^2] = \sum_{s \in \mathcal{S}_i} \mathbb{E}[X_i^2|s]\mathbb{P}(s)$$

The conditional expectation term is distributed as a geometric variable, which allows us to apply the following bound.

$$\begin{aligned} \mathbb{E}[X_i^2|s] &\leq \frac{2}{\left(1 - \sum_{j=1}^{i-1} q_j\right)^2} \\ &\leq \frac{2}{\left(1 - \frac{i-1}{N}\alpha_{max}\right)^2} \end{aligned} \tag{3.10}$$

We can now obtain a bound for variance by substituting (3.10) into the variance expression:

$$Var[\theta(\infty)] \leq \sum_{i=0}^{m-1} \frac{2}{(1 - \frac{i}{N}\alpha_{max})^2}$$

As N grows large, the $\frac{i}{N}$ step becomes finer and hence can be approximated using an integral. Using the definition that $m = 2\beta N$, we can write:

$$\begin{aligned} \sum_{i=0}^{m-1} \frac{2}{(1 - \frac{i}{N}\alpha_{max})^2} &\approx N \int_0^{2\beta} \frac{2}{(1 - \alpha_{max}x)^2} dx \\ &= N\Big[\frac{2}{\alpha_{max}(1 - 2\beta\alpha_{max})} - 2\Big] \\ &\leq \frac{2N}{\alpha_{max}(1 - 2\beta\alpha_{max})} \end{aligned}$$

Finally, we can use the above result to prove concentration using Chebyshev's inequality:

$$\mathbb{P}\Big(|\theta(\infty) - \mathbb{E}[\theta(\infty)]| \geq \frac{1}{N^{0.5+\epsilon}}\Big) \leq \frac{2N^{-2\epsilon}}{\alpha_{max}(1 - 2\alpha_{max}\beta)}$$

□

3.4 Theorem 2 Proof

We have proven all preliminary lemmas for Theorem 2. We will now formally prove Theorem 2. Throughout this proof, we make consistent references to the bounds established in Lemma 2. We will thus define B_L and B_U to represent the lower and upper bounds, respectively. Namely:

$$B_L = \mathbb{E}[\theta(\infty)] - \frac{1}{N^{0.5+\epsilon}}$$
$$B_U = \mathbb{E}[\theta(\infty)] + \frac{1}{N^{0.5+\epsilon}}$$

Using the fact that $|y_{i,\theta(t)}(t) - y^*_{i,\theta(t)}| \leq 2, \forall i = 1, 2$ paired with the result of Lemma 1 and Jensen's inequality, we deduce:

$$\begin{aligned}
\mathbb{E}\big[|h(\infty) - h^*|\big] &\leq 2\sum_{i=1}^{2} \mathbb{E}\big[|y_{i,\theta(\infty)}(t) - y^*_{i,\theta(\infty)}|\big] \\
&\leq 2\sum_{i=1}^{2} \sqrt{\mathbb{E}\big[(y_{i,\theta(\infty)}(t) - y^*_{i,\theta(\infty)})^2\big]} \\
&\leq 2\sum_{i=1}^{2} \sqrt{\mathbb{E}[(y_{i,\theta(\infty)}(t) - y^*_{i,\theta(\infty)})^2 \mathbb{1}_{\theta(\infty)\in\mathcal{B}}] + 4\mathbb{P}\big(\theta(\infty) \notin \mathcal{B}\big)}
\end{aligned} \tag{3.11}$$

where $\mathcal{B}$ is defined as the interval $[B_L, B_U]$. We thus want to show that the right-hand side of (3.11) is small. Note that the final line follows from the fact that we can partition and bound the expectation term on whether or not $\theta(\infty)$ lies in the concentration. From Lemma 2, we know that the $\mathbb{P}(\theta(\infty) \notin [B_L, B_U])$ is small. We can similarly apply the results of Theorem 1 to prove the first term is bounded.

$$\mathbb{E}[(y_{i,\theta(\infty)}(t) - y^*_{i,\theta(\infty)})^2 \mathbb{1}_{\theta(\infty)\in[B_L,B_U]}] \leq \max_{b\in[B_L,B_U]} \mathbb{E}[(y_{i,b}(t) - y^*_{i,b})^2]$$

We can thus apply the union bound for all b to upper bound the max expression. Note that the quantity in the expectation is precisely bounded from Theorem 1.

$$\mathbb{E}[(y_{i,\theta(\infty)}(t) - y^*_{i,\theta(\infty)})^2 \mathbb{1}_{\theta(\infty)\in[B_L,B_U]}] \leq \big|\{b : b \in [B_L, B_U]\}\big| \frac{6}{N\alpha_i}$$

Substituting this result into (3.11):

$$\mathbb{E}\big[|h(\infty) - h^*|\big] \leq 2\sum_{i=1}^{2}\sqrt{\big|\{b : b \in [B_L, B_U]\}\big|\frac{6}{N\alpha_i} + 4\mathbb{P}\big(\theta(\infty) \notin [B_L, B_U]\big)} \\ \leq 4\sqrt{\big|\{b : b \in [B_L, B_U]\}\big|\frac{6}{N\alpha_{min}} + 4\mathbb{P}\big(\theta(\infty) \notin [B_L, B_U]\big)} \tag{3.12}$$

where $\alpha_{min} = \min(\alpha_1, \alpha_2)$. We thus see that the two terms in (3.12) can be thought of distinctly as a union bound result from Theorem 1 along with a concentration result from the coupon collector problem. To arrive at the aforementioned bound, we can directly apply Lemma 2 with the values $\epsilon = \frac{1}{6}$ and $|B_L - B_U| = N^{0.5+\epsilon}$. With these values, it follows directly that

$$\mathbb{E}\big[\big|h(\infty) - h^*\big|\big] \leq O\Big(\frac{1}{N^{\frac{1}{12}}}\Big)$$

We have thus demonstrated that the true LRU hit rate converges to the TTL hit rate approximation.

CHAPTER 4

EXTENSION TO SEGMENTED LRU

After demonstrating convergence in LRU, we will now extend our model to LRU(m). Recall that LRU(m) is a partitioned version of LRU where items move from list to list depending on the frequency of item requests. As items are more frequently requested, they move to higher, more protected lists. For consistent notation throughout this chapter, we will say that all items enter list 1, which represents the lowest priority list. If an item is requested that is currently in list i, the item is moved to the head of list $\max(m, i+1)$. Thus, higher list numbers represent more popular items.

In this chapter, we first present an alternative Markov chain that inherently puts a larger emphasis on the structure of the cache rather than only considering request times. The key benefit of this framework is its adaptability to most LRU variants. We will first present an alternate model for LRU, and then generalize this model to LRU(m).

4.1 Alternate LRU Model

The key aspect of this model is that we are looking at elements in the cache, instead of request times. Consider a population as described in Section 2.1, where there are two types of distinct objects with N items per type. Now, consider a parameter β_0 which is defined as a fractional component that signifies the size of the cache with respect to the size of the population. Alternatively, we could define the size of the cache to be m, and define $m = 2N\beta_0$. The cache is thus composed of m, $\frac{1}{2N}$ slots since $\beta_0 = \frac{m}{2N}$.

We now define the state variable, $y_{i,\beta}(t)$, which represents the fraction of type i items in the first $2\beta N$ positions in the cache. Note that β is always measured from the head of the cache, or the item with the highest priority. Thus, for LRU(m), the state variable will always count type i items starting

at the highest priority list and move toward the tail. Note $\beta \in \{\frac{1}{2N}, \ldots, \beta_0\}$.

For ease of notation, we now define two regions of the cache that are dependent on the state variable. Define the exclude region as the elements that are currently in the cache but are not considered in the state variable $y_{i,\beta}$ for the given β. The exclude region is thus the collection of items with $\beta \in \{\beta + \frac{1}{2N}, \ldots, \beta_0\}$. Conversely, define the include region as the collection of elements that are currently in the cache and have $\beta \in \{0, \frac{1}{2N}, \ldots, \beta\}$.

We will now define the model for LRU. Observe the defined Markov chain for $y_{1,\beta}$ below.

$$y_{1,\beta}(t + \frac{1}{N}) =$$
$$\begin{cases} y_{1,\beta}(t) + \frac{1}{N} - \tilde{y}_{1,\beta}(t) & \text{w.p.} \quad \alpha_1(y_{1,\beta_0}(t) - y_{1,\beta}(t)) & \text{Type 1 Hit Exclude} \\ y_{1,\beta}(t) - \tilde{y}_{1,\beta}(t) & \text{w.p.} \quad \alpha_2(y_{2,\beta_0}(t) - y_{2,\beta}(t)) & \text{Type 2 Hit Exclude} \\ y_{1,\beta}(t) + \frac{1}{N} - \tilde{y}_{1,\beta}(t) & \text{w.p.} \quad \alpha_1(1 - y_{1,\beta_0}(t)) & \text{Type 1 Miss} \\ y_{1,\beta}(t) - \tilde{y}_{1,\beta}(t) & \text{w.p.} \quad a_2(1 - y_{2,\beta_0}(t)) & \text{Type 2 Miss} \\ y_{1,\beta}(t) & \text{w.p.} \quad \alpha_1 y_{1,\beta}(t) & \text{Type 1 Hit Include} \\ y_{1,\beta}(t) & \text{w.p.} \quad \alpha_2 y_{2,\beta}(t) & \text{Type 2 Hit Include} \end{cases} \tag{4.1}$$

Although seemingly more complex than the model presented in (2.2), this model is quite intuitive based on the definitions of the include and exclude zone. To begin, observe that $y_{i,\beta_0}(t)$ represents the total fraction of type i items in the cache at time t. We can therefore fully define all states of the Markov chain just as before. We can also derive a differential equation to approximate this system using the conditional expectation of the drift term. To begin:

$$\mathbb{E}\big[y_{1,\beta}(t + \frac{1}{N}) - y_{1,\beta}(t)\big] = 0 \tag{4.2}$$

Using this Markov chain, we can then derive a differential equation:

$$\mathbb{E}\Big[\mathbb{E}\Big[y_{1,\beta}(t+\frac{1}{N}) - y_{1,\beta}(t)\Big]\Big|y(t)\Big] = \mathbb{E}\big[\frac{\alpha_1}{N}(1-y_{1,\beta}(t)) - (1-\alpha_1 y_{1,\beta}(t) - \alpha_2 y_{2,\beta}(t))\tilde{y}_{1,\beta}(t)\big] \tag{4.3}$$

Taking $N \to \infty$ we obtain the following partial differential equation.

$$\frac{\partial y_{1,\beta}(t)}{\partial t} = \alpha_1(1 - y_{1,\beta}(t)) - (1 - \alpha_1 y_{1,\beta}(t) - \alpha_2 y_{2,\beta}(t))\frac{\partial y_{1,\beta}(t)}{\partial \beta} \tag{4.4}$$

By definition, (4.4) can be reduced to an equation of two variables using the following relationship:

$$y_{1,\beta}(t) + y_{2,\beta}(t) = \beta \tag{4.5}$$

We now get a partial differential equation of only $y_{1,\beta}$ and β.

$$\frac{\partial y_{1,\beta}(t)}{\partial t} = \alpha_1(1 - y_{1,\beta}(t)) - (1 - \alpha_1 y_{1,\beta}(t) - \alpha_2(\beta - y_{1,\beta}(t)))\frac{\partial y_{1,\beta}(t)}{\partial \beta} \tag{4.6}$$

In steady state, we therefore get the following ordinary differential equation.

$$\frac{dy_{1,\beta}}{d\beta} = \frac{\alpha_1(1 - y_{1,\beta})}{1 - \alpha_1 y_{1,\beta} - \alpha_2(\beta - y_{1,\beta})} \tag{4.7}$$

4.2 LRU(2) Extension

We now extend this model to LRU(2). LRU(2) has inherently different dynamics compared to LRU. We propose two Markov chains to encompass the dynamics of the entire cache. For this analysis, we assume that both caches are of equal size. We define cache 2 as $\beta \in \{\frac{1}{2N}, \ldots, \frac{\beta_0}{2}\}$ and cache 1 as $\beta \in \{\frac{\beta_0}{2} + \frac{1}{2N}, \ldots, \beta_0\}$. The assumption that both cache 1 and cache 2 have equal size can be easily relaxed by adjusting the β limits for each list. Also, note that the model can be directly generalized to more than two lists, by adding Markov chains for different regions of the cache.

We first define the model for $\beta \in \{\frac{\beta_0}{2} + \frac{1}{2N}, \ldots, \beta_0\}$ in the following Markov chain.

$$y_{1,\beta}(t + \frac{1}{N}) =$$

$$\begin{cases} y_{1,\beta}(t) + \frac{1}{N} - \tilde{y}_{1,\beta}(t) & \text{w.p.} \quad a_1(y_{1,\beta_0}(t) - y_{1,\beta}(t)) & \text{Type 1 Hit Exclude} \\ y_{1,\beta}(t) - \tilde{y}_{1,\beta}(t) & \text{w.p.} \quad a_2(y_{2,\beta_0}(t) - y_{2,\beta}(t)) & \text{Type 2 Hit Exclude} \\ y_{1,\beta}(t) + \frac{1}{N} - \tilde{y}_{1,\beta}(t) & \text{w.p.} \quad a_1(1 - y_{1,\beta_0}(t)) & \text{Type 1 Miss} \\ y_{1,\beta}(t) - \tilde{y}_{1,\beta}(k) & \text{w.p.} \quad a_2(1 - y_{2,\beta_0}(t)) & \text{Type 2 Miss} \\ y_{1,\beta}(t) & \text{w.p.} \quad a_1 y_{1,\beta}(t) & \text{Type 1 Hit Include} \\ y_{1,\beta}(t) & \text{w.p.} \quad a_2 y_{2,\beta}(t) & \text{Type 2 Hit Include} \end{cases} \tag{4.8}$$

This model is identical of that proposed in (4.1). Going through the same

analysis that was presented in (4.2) - (4.6), we obtain the following ordinary differential equation.

$$\frac{dy_{1,\beta}}{d\beta} = \frac{\alpha_1(1 - y_{1,\beta})}{1 - \alpha_1 y_{1,\beta} - \alpha_2(\beta - y_{1,\beta})} \tag{4.9}$$

Similarly, we now define the model for $\beta \in \{\frac{1}{2N}, \ldots, \frac{\beta_0}{2}\}$ in the following Markov chain.

$$y_{1,\beta}(t + \frac{1}{N}) =$$

$$\begin{cases} y_{1,\beta}(t) + \frac{1}{N} - \tilde{y}_{1,\beta}(t) & \text{w.p.} \quad a_1(y_{1,\beta_0}(t) - y_{1,\beta}(t)) & \text{Type 1 Hit Exclude} \\ y_{1,\beta}(t) - \tilde{y}_{1,\beta}(t) & \text{w.p.} \quad a_2(y_{2,\beta_0}(t) - y_{2,\beta}(t)) & \text{Type 2 Hit Exclude} \\ y_{1,\beta}(t) & \text{w.p.} \quad a_1(1 - y_{1,\beta_0}(t)) & \text{Type 1 Miss} \\ y_{1,\beta}(t) & \text{w.p.} \quad a_2(1 - y_{2,\beta_0}(t)) & \text{Type 2 Miss} \\ y_{1,\beta}(t) & \text{w.p.} \quad a_1 y_{1,\beta}(t) & \text{Type 1 Hit Include} \\ y_{1,\beta}(t) & \text{w.p.} \quad a_2 y_{2,\beta}(t) & \text{Type 2 Hit Include} \end{cases} \tag{4.10}$$

This model is quite similar to that proposed in (4.1). The key difference between the two models arises in the dynamics of an entire cache miss. Because cache 2 only receives objects when there is a hit in cache 1, there is no change in the state of the system. We can therefore perform identical analysis as above to arrive at an approximate differential equation. The steady state differential equation for this system is:

$$\frac{dy_{1,\beta}}{d\beta} = \frac{\alpha_1(y_{1,\beta_0} - y_{1,\beta})}{\alpha_1(y_{1,\beta_0} - y_{1,\beta}) + \alpha_2(y_{2,\beta_0} - y_{2,\beta})} \tag{4.11}$$

Interestingly, (4.7), (4.9), and (4.11) all have the same general form with their ODE solution. Thus, once we can demonstrate convergence for LRU, we will be able to directly extend the result to LRU(m). In [6], Gast and Van Houdt demonstrated that the characteristic time, T_i, for each list, i, is defined precisely as the fixed point of:

$$m_l = N \sum_{k=1}^{2} \frac{\left(e^{\alpha_k \cdot T_1} - 1\right) \ldots \left(e^{\alpha_k \cdot T_l} - 1\right)}{1 + \sum_{j=1}^{h} \left(e^{\alpha_k \cdot T_1} - 1\right) \ldots \left(e^{\alpha_k \cdot T_j - 1}\right)} \tag{4.12}$$

where m_l is the size of the l^{th} list, and h is the number of lists. We have

previously demonstrated that the steady state solution of the differential equations for our model indeed satisfies (4.12), but it has proven difficult to achieve a bound result as we initially did in Theorem 1. The solution to these ordinary differential equations has an implicit form which has made the proof method presented in the previous chapter difficult to apply. We continue to seek viable Lyapunov functions and other techniques to prove this model.

CHAPTER 5

SIMULATION

In this chapter, we aim to demonstrate steady state convergence of our TTL approximation to the given model. The hit rate metric will be used to compare different models. For this simulation, define I as the number of iterations performed. Let Z_i to be an indicator variable defined as follows:

$$Z_i = \begin{cases} 1 & \text{If the } i^{th} \text{ request results in a hit} \\ 0 & \text{else} \end{cases}$$

Using this definition, define the empirical hit rate, $\hat{h}(i)$, as follows:

$$\hat{h}(i)\frac{\sum_{j=1}^{i} Z_j}{i}, \quad i \in \{1, 2, \ldots, I\} \tag{5.1}$$

We will now present simulation results for LRU and segmented LRU. The simulations are performed identically given the following procedure. Initially, items are randomly selected until the cache holds m distinct objects. We will define this process as the initialization of the cache. We then randomly sample the probability distribution defined in (2.1) and simulate the dynamics of the cache according to the caching algorithm for each $i \in \{1, 2, \ldots, I\}$. At each iteration, the empirical hit rate is recorded. The TTL hit rate will be calculated at the end of the simulation. Throughout this chapter, we will be comparing the TTL approximation presented previously to the empirical hit rate defined in (5.1).

5.1 LRU Results

In this section, we present LRU hit rate results over a collection of test settings. We will be testing the model presented in Section 2.1. For the sake of presentation, we will maintain all conditions while varying only the cache

size. Cache size was chosen to be the independent variable because in many practical situations, the cache designer may only have the freedom to control the caching algorithm and cache size. Thus, we are demonstrating how hit rate is affected by cache size under the following conditions:

- $I = 75,000$ iterations
- Population Size $= 10,000$
- $\alpha_1 = 0.25$

To begin, in Figure 5.1, we plot the hit rate compared to the iteration count for the conditions described above. This plot can be thought of as a sample trajectory for the hit rate as it tends to its steady state.

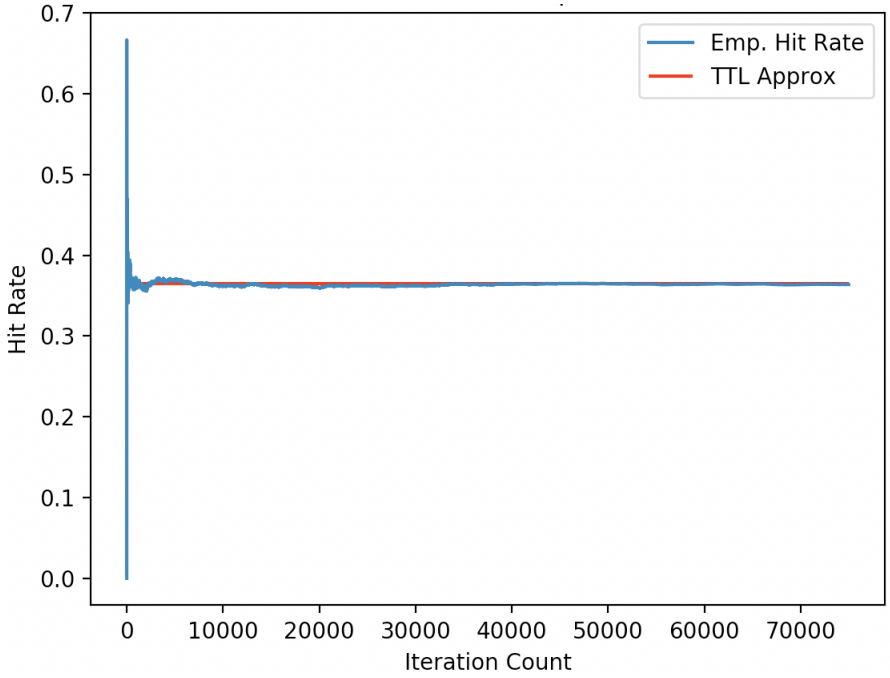

Figure 5.1: Example LRU Hit Rate Plot

Note that this plot was generated assuming a cache size of $3,000$ items. There are two distinct regions in this figure. Specifically, we see that for approximately the first $20,000$ iterations, the simulation is in a transient, mixing state. After this point, the Markov chain appears to obtain a steady state distribution for the hit rate. The TTL approximation, which is constant across all iterations, qualitatively matches the steady state distribution well. We will now quantitatively compare hit rate results for both the simulation

and approximation for varying cache sizes in Table 5.1. To obtain a better approximation of the steady state distribution, note that the empirical hit rate, $\hat{h}$, was calculated using the mean of the final $10,000$ samples of the simulated hit rate.

Table 5.1: LRU Hit Rate Comparison for Varying Cache Sizes

Cache Size, [# Items]	Empirical Hit Rate, $\hat{h}$	TTL Hit Rate, h^*	Error [%]
$1,000$	0.123	0.124	0.819
$2,000$	0.245	0.246	0.131
$3,000$	0.362	0.365	0.777
$4,000$	0.481	0.480	0.225
$5,000$	0.591	0.591	0.086
$6,000$	0.694	0.696	0.303
$7,000$	0.792	0.792	0.132

Through this simulation, we observe that the true empirical hit rate is within 1% of the TTL approximation for all trials. It is thus apparent that the proposed approximation fits the LRU model presented in (2.2) well.

5.2 LRU(2) Results

In this section, we present simulation LRU(2) hit rate results. We will be testing the model presented in Section 4.2. We will once again keep all simulation parameters constant with the exception of cache size. Furthermore, all simulations will be performed under the exact conditions that were presented in Section 5.1, with the exception that the cache is now partitioned into two, equally sized subcaches. The dynamics of the cache will be governed by LRU(2). We will thus have a direct comparison between LRU and LRU(2).

We begin analysis by providing a sample trajectory of hit rate compared to iteration count in Figure 5.2.

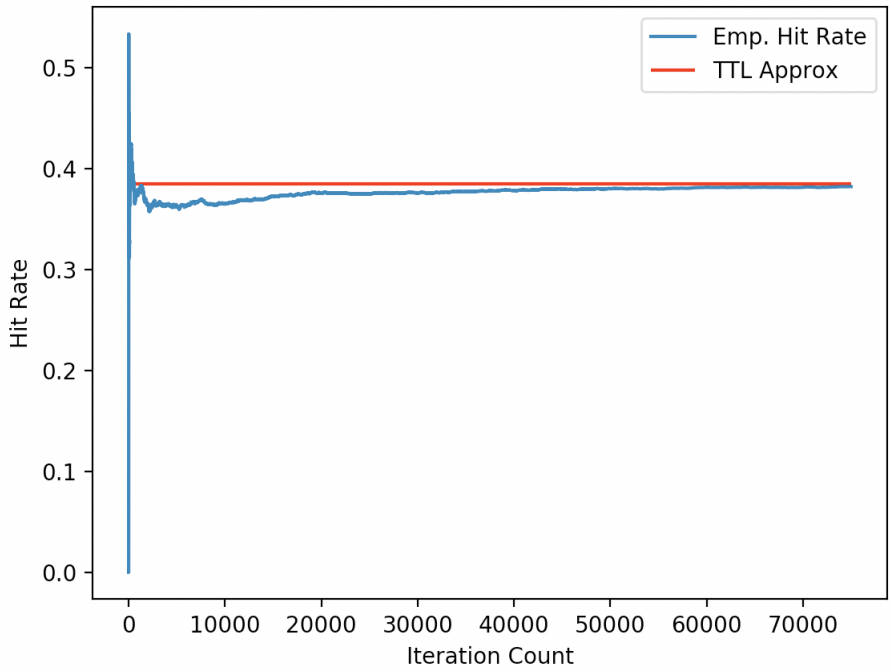

Figure 5.2: Example LRU(2) Hit Rate Plot

From this plot, it is evident that the hit rate asymptotically increases to the proposed TTL approximation. In this situation, we observe slight performance benefits by partitioning the cache. We will now quantitatively compare hit rate results for both the simulation and approximation for varying cache sizes. The results are tabulated in Table 5.2. Note that all of these simulations were performed such that the size of each sub cache was exactly half of the total cache.

Table 5.2: LRU(2) Hit Rate Comparison for Varying Cache Sizes

Cache Size [# Items]	Empirical Hit Rate, $\hat{h}$	TTL Hit Rate, h^*	Error [%]
$1,000$	0.132	0.132	0.455
$2,000$	0.262	0.261	0.41
$3,000$	0.385	0.385	0.042
$4,000$	0.501	0.503	0.31
$5,000$	0.608	0.614	0.905
$6,000$	0.707	0.715	1.145
$7,000$	0.804	0.805	0.214

We thus notice that the provided TTL approximation for segmented LRU

given in (4.12) also closely fits the model proposed in Section 4.2. Further, notice that both the empirical and TTL hit rates are consistently higher for segmented LRU compared to LRU. On average, LRU(2) provided a 1.46% empirical hit rate improvement compared to LRU. It may be hypothesized that increasing partitions should always lead to higher hit rates; however, this may not always be the case. This observation leads to the interesting question of optimal partitioning schemes, and whether the TTL approximation can predict this. We will present some speculations regarding these simulations in Chapter 6.

CHAPTER 6

VARIANCE CORRECTED APPROXIMATION

In this chapter, we motivate and present a continuous time model that parallels our LRU model.

6.1 TTL Correction Motivation

While simulating the models in Section 4.2, it was observed that adding too many partitions to a cache governed by LRU(m) would decrease performance. Intuitively, this observation makes sense. For example, suppose there was a cache of size m that was composed of m, one-item subcaches. In this scenario, the only way an item could move from the first list to any higher list would be if successive requests occurred for the same item. Given that the system has sufficiently many objects, this event is unlikely to occur.

Furthermore, we observed that as the number of partitions increased, the TTL approximation began to separate from the simulation hit rate. This phenomenon was especially observed when the cache size was small compared to the population size, resulting in low hit rates. For example, consider Figure 6.1 that plots the number of subcaches compared to the hit rate. Note that this plot was generated using a population size of $5,000$ items and a 100 item cache size.

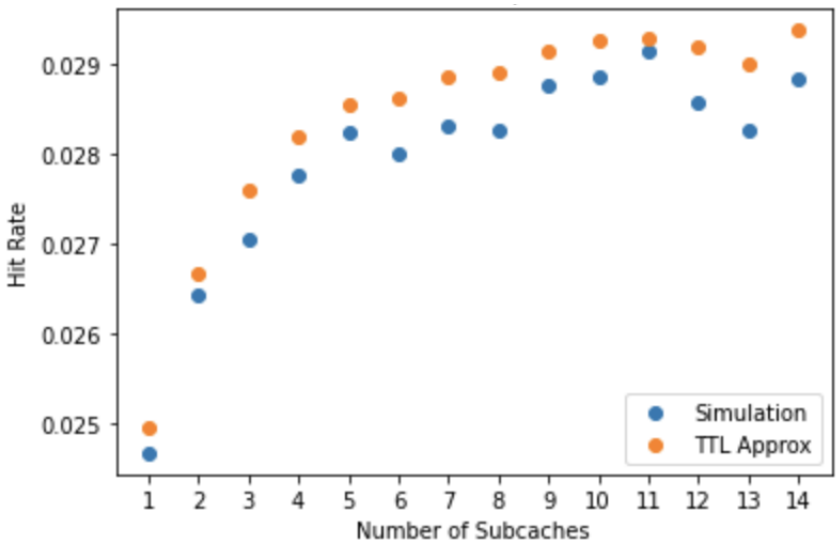

Figure 6.1: Example LRU(2) Hit Rate Plot

We therefore hypothesize the need for a variance correcting term. The TTL approximation is deterministic in nature and thus constant over time. By adding a variance term, we add randomness to the approximation with the thought that this randomness will better fit the system over time.

6.2 Variance Corrected Approximation

In order to model this variance correction term, we will transition the model presented in (4.1) to resemble a Poisson arrival process. To begin, consider two independent Poisson arrival processes that represent the arrival of a type i request:

- Type 1 Arrivals $\sim Poi(N\lambda_1)$
- Type 2 Arrivals $\sim Poi(N\lambda_2)$

where λ_i is the rate of the process. From the properties of independent Poisson processes, these two arrival rates can be summed in order to define the overall arrival process rate as $Poi(N(\lambda_1 + \lambda_2))$. Hence, given an arrival, the probability that the item is type i is:

$$\frac{\lambda_i}{\lambda_1 + \lambda_2}$$

From the nature of the Poisson arrival process, the probability that there is an arrival in $[t, t+\delta]$ is the arrival rate multiplied by δ. We can now define the Markov chain.

$$y_{1,\beta}(t+\delta) =$$

$$\begin{cases} y_{1,\beta}(t) + \frac{1}{N} - \tilde{y}_{1,\beta}(t) & \text{w.p.} \quad \delta\lambda_1 N * (y_{1,\beta_0}(t) - y_{1,\beta}(t)) & \text{Type 1 Hit Exclude} \\ y_{1,\beta}(t) - \tilde{y}_{1,\beta}(t) & \text{w.p.} \quad \delta\lambda_2 N * (y_{2,\beta_0}(k) - y_{2,\beta}(t)) & \text{Type 2 Hit Exclude} \\ y_{1,\beta}(t) + \frac{1}{N} - \tilde{y}_{1,\beta}(t) & \text{w.p.} \quad \delta\lambda_1 N * (1 - y_{1,\beta_0}(t)) & \text{Type 1 Miss} \\ y_{1,\beta}(t) - \tilde{y}_{1,\beta}(t) & \text{w.p.} \quad \delta\lambda_2 N * (1 - y_{2,\beta_0}(t)) & \text{Type 2 Miss} \\ y_{1,\beta}(t) & \text{w.p.} \quad \delta\lambda_1 N * y_{1,\beta}(t) & \text{Type 1 Hit Include} \\ y_{1,\beta}(t) & \text{w.p.} \quad \delta\lambda_2 N * y_{2,\beta}(t) & \text{Type 2 Hit Include} \\ y_{1,\beta}(t) & \text{w.p.} \quad 1 - N\delta(\lambda_1 + \lambda_2) & \text{No Arrival} \end{cases} \tag{6.1}$$

Observe the clear similarities between the above expression and (4.1), with the primary difference being that there is a non-zero probability that there is no arrival in the δ interval. Given this model, our objective is to define a stochastic difference equation. We will therefore develop an expression for $y_{1,\beta}(t+\delta) - y_{1,\beta}(t)$. To develop the variance term, we will represent the probabilities of each event as the sum of Bernoulli random variables:

$$\begin{aligned} y_{1,\beta}(t+\delta) - y_{1,\beta}(t) = \big[\frac{1}{N} - \tilde{y}_{1,\beta}(t)\big]\Big[\sum_{i=1}^{N(1-y_{1,\beta}(t))} Ber(\lambda_1\delta)\Big] \\ - \tilde{y}_{1,\beta}(t)\Big[\sum_{j=1}^{N(1-y_{2,\beta}(t))} Ber(\lambda_2\delta)\Big] \end{aligned} \tag{6.2}$$

This can be expressed as a mean plus variance equation though the central limit theorem. As an example, observe:

$$\begin{aligned} \sum_{j=1}^{n(1-y_{2,\beta}(t))} Ber(\lambda_2\delta) &\to \sqrt{n}\Big(\lambda_2\delta\sqrt{n}(1-y_{2,\beta}(t)) + \mathcal{N}\big(0,\ \lambda_2\delta(1-y_{2,\beta}(t))(1-\lambda_2\delta)\big)\Big) \\ &\to \sqrt{n}\Big(\lambda_2\delta\sqrt{n}(1-y_{2,\beta}(t)) + \mathcal{N}\big(0,\ \lambda_2\delta(1-y_{2,\beta}(t))\big)\Big) \\ &\to n\lambda_2\delta(1-y_{2,\beta}(t)) + \mathcal{N}\big(0,\ n\lambda_2\delta(1-y_{2,\beta}(t)\big) \end{aligned}$$

where the number of type i items, N, is denoted as n to avoid confusion with the normal distribution, $\mathcal{N}$. Further, we can make the following approximation:

$$\tilde{y}_{1,\beta} = y_{1,\beta} - y_{1,\beta-1/n} \approx \frac{1}{n}\frac{\partial y_{1,\beta}}{\partial \beta}$$

Now, the SDE is achieved by combining the above result with (6.2).

$$\begin{aligned} dy_{1,\beta} = (1 - \frac{\partial y_{1,\beta}}{\partial \beta})\Big[\lambda_1(1 - y_{1,\beta})dt + \frac{1}{\sqrt{n}}\sqrt{\lambda_1(1 - y_{1,\beta})}dw_1\Big] \\ - \frac{\partial y_{1,\beta}}{\partial \beta}\Big[\lambda_2(1 - y_{2,\beta})dt + \frac{1}{\sqrt{n}}\sqrt{\lambda_2(1 - y_{2,\beta})}dw_2\Big] \end{aligned}$$

In future work, we plan on developing this SDE and solving in order to more accurately fit the model. It remains speculative how this model will fit the system, and whether it will be more accurate than the deterministic TTL approximation.

CHAPTER 7

CONCLUSION

In this thesis, we analyzed the least recently used caching algorithm. We proposed a Lyapunov based proof to demonstrate the convergence of LRU hit rate to the fixed point of a differential equation, known as the TTL approximation. We then expanded this Markov model to a general form that can accommodate both LRU and LRU(m). A Lyapunov based proof of the convergence of the new model to the TTL approximation remains an open question. We provided simulation results for both algorithms to demonstrate the validity of both TTL approximations. From these simulations, we observed a possible shortcoming of the current approximation in situations where the cache size is small compared to the object population. Our current focus resides in further developing a corrected form for this observation using a variance corrected expression.

REFERENCES

[1] R. Fagin, "Asymptotic miss ratios over independent references," *Journal of Computer and System Sciences*, vol. 14, no. 2, pp. 222 – 250, 1977. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0022000077800147

[2] H. Che, Y. Tung, and Z. Wang, "Hierarchical web caching systems: Modeling, design and experimental results," *Selected Areas in Communications, IEEE Journal on*, vol. 20, pp. 1305 – 1314, 10 2002.

[3] C. Fricker, P. Robert, and J. Roberts, "A versatile and accurate approximation for LRU cache performance," *CoRR*, vol. abs/1202.3974, 2012. [Online]. Available: http://arxiv.org/abs/1202.3974

[4] R. Karedla, J. S. Love, and B. G. Wherry, "Caching strategies to improve disk system performance," *Computer*, vol. 27, no. 3, pp. 38–46, 1994.

[5] Q. Huang, K. Birman, R. van Renesse, W. Lloyd, S. Kumar, and H. C. Li, "An analysis of FaceBook photo caching," in *Proceedings of the Twenty-Fourth ACM Symposium on Operating Systems Principles*, ser. SOSP '13. New York, NY, USA: Association for Computing Machinery, 2013. [Online]. Available: https://doi.org/10.1145/2517349.2522722 p. 167–181.

[6] N. Gast and B. Van Houdt, "TTL approximations of the cache replacement algorithms LRU(m) and h-LRU," *Performance Evaluation*, Sep. 2017. [Online]. Available: https://hal.inria.fr/hal-01622059

[7] P. Flajolet, D. Gardy, and L. Thimonier, "Birthday paradox, coupon collectors, caching algorithms and self-organizing search," *Discret. Appl. Math.*, vol. 39, pp. 207–229, 1992.