Non-Uniform Balls-into-Bins and the Analysis of Lock-Free Algorithms[1]

Dan Alistarh
Microsoft Research

Thomas Sauerwald
University of Cambridge

Milan Vojnovic
Microsoft Research

Technical Report
MSR-TR-2014-153

Microsoft Research
Microsoft Corporation
One Microsoft Way
Redmond, WA 98052
http://www.research.microsoft.com

[1]Dan Alistarh is with Microsoft Research, Cambridge, United Kingdom (daalista@microsoft.com). Thomas Sauerwald is with the University of Cambridge (tms42@cam.ac.uk). Milan Vojnović is with Microsoft Research, Cambridge, United Kingdom (milanv@microsoft.com).

Abstract – In this work, we consider the following random process, motivated by the analysis of lock-free concurrent algorithms under stochastic schedulers. In each step, a new ball is allocated into one of n bins, according to a distribution $\mathbf{p} = (p_1, p_2, \ldots, p_n)$, where each ball goes to bin i with probability p_i. When some bin first reaches a set threshold of balls, it registers a *win*, and resets its ball count to 1. At the same time, bins whose ball count was close to the threshold also get reset on a win, to 0 balls, being penalized for *almost* winning. We are interested in two questions: how often does *some* bin win (*system latency*), and how often does a *specific* bin win (*individual latency*)?

We provide asymptotically tight bounds for the system and individual latency of this general concurrency pattern, for arbitrary scheduling distributions $\mathbf{p}$. We find that, surprisingly, a simple characterization exists: in expectation, the system will complete a new operation every $\Theta(1/\|\mathbf{p}\|_2)$ steps, while thread i will complete a new operation every $\Theta(\|\mathbf{p}\|_2/p_i^2)$ steps. The proof of this result is quite involved, as it requires a careful analysis of how the higher norms of the vector $\mathbf{p}$ influence the occupancy distribution of bins and latencies in this random process. Our result offers a simple relation between the scheduling distribution and the average performance of concurrent algorithms, which has several practical applications.

1 Introduction

The Problem. In this paper, we consider the following stochastic process: we are given n bins, a probability distribution $\mathbf{p} = (p_1, p_2, \ldots, p_n)$, and a constant $k \geq 2$. In each step, we allocate one new ball into one of the bins, according to the distribution $\mathbf{p}$, where bin i gets the ball with probability p_i. Bins continue to acquire balls, until one of them first reaches a threshold of $k + 1$ balls. At this time, the corresponding bin registers a *win*, and gets reset to containing a single ball. Moreover, bins which contain k balls at this time get *reset* to having 0 balls, being penalized for *almost* winning. All other bins maintain their ball count. Given such a process, we are interested in two questions: how often does *some bin* register a win, and how often does *a specific bin* i register a win?

Background. The motivation for this process is the scheduling of *conflicting* concurrent operations in a distributed system, such as asynchronous shared-memory [17] or concurrent databases [22]. The immediate application, and the inspiration for the formulation above, is the performance analysis of *lock-free (non-blocking)* concurrent algorithms [14]: each bin represents an execution thread, with a set of data structure operations to perform, and each operation is composed of a constant number of basic steps. We are interested in how often, on average, the system completes a new operation (*system latency*), and a specific thread i completes a new operation (*individual latency*).

Lock-free algorithms usually follow a specific *scan-and-validate* pattern, consisting of a *scan* region designed to assess the state of the data structure and to locally compute an updated state, and ending with *validate* step consisting of a read-modify-write operation which atomically checks that the state of the data structure is still consistent with the scan, and, if so, commits the updated state. Crucially, this last validation step may fail if the data structure state changes between the scan and the validation step, meaning that some other operation succeeded in the meantime. If validation fails, the current operation has to restart. In the process described above, k is the length of an operation. The winning ball threshold represents the number of steps needed to complete an operation. It is offset by 1 to model the fact that processes failing their validation step incur this as an extra cost before re-issuing their operation.

The key property of this algorithmic pattern is that an operation fails only if a concurrent one succeeds: thus, the system as a whole is guaranteed to progress, although individual processes may, in theory, starve. Traditionally, this has led lock-free algorithms to be less studied by the theory community, although they are very popular among practitioners due to their relatively simple structure and excellent practical performance, e.g., [9, 14, 15, 18, 23, 25]. Herlihy and Shavit [15] suggest that a reason for this popularity is that, on realistic schedules, lock-free algorithms ensure that each operation completes in a finite number of its own steps, i.e., they are in fact *wait-free* [12].

Recently, Alistarh, Censor-Hillel, and Shavit [1] provided a simple stochastic condition on schedules, which is enough to ensure that a lock-free algorithm becomes wait-free with probability 1: roughly, it is enough for the schedule to have some non-zero probability of picking each thread i in each step. They also gave evidence that such schedules are common in practice, and showed that it is possible to characterize the performance for a general family of lock-free algorithms, called *Single-CAS Universal (SCU)*,[1] under an idealized *uniform stochastic scheduler*, which picks

[1]The SCU algorithmic pattern is *lock-free universal* [1, 12] in the sense that it can be used to provide a lock-free concurrent implementation for every object that can be specified sequentially.

the next processor to schedule in each step with uniform probability. Practically, their analysis gave tight latency bounds for the process described above, in the uniform case $\mathbf{p} = (1/n, 1/n, \ldots, 1/n)$.

The Question. While reference [1] gave a first way of performing average-case analysis for concurrent algorithms under a uniform scheduler, it leaves open the question of what happens under non-uniform ones. (Unfortunately, their technique hinges on the symmetry of the distribution, and breaks down at the first step for non-uniform distributions.)

Bounding latencies for general $\mathbf{p}$ is important for several reasons. First, many modern micro-processor architectures exhibit consistently non-uniform scheduling distributions, since memory access times for different threads are inherently heterogeneous. (Please see Appendix F or references [7, 16] for examples.) It is not clear how non-uniformity affects the accuracy of the average-case analysis for lock-free algorithms. Further, a number of interesting questions remain open: what is the relative *performance advantage* of a thread that gets scheduled with higher probability? What is the *worst-case* scheduling distribution for lock-free algorithms in terms of system latency? What is the *optimal* system latency achievable, while still guaranteeing some minimal progress guarantees for individual threads? This last question is especially important, since it is common for programmers to alter the scheduling distribution using back-offs; however, determining the optimal back-off scheme for a given set of latency constraints is still an open problem.

Contribution. In this paper, we give asymptotically tight bounds on the performance of lock-free algorithms following the universal lock-free *SCU* pattern, in the common case where operations have constant length, with respect to an arbitrary sequence of scheduling distributions $\mathbf{p}(n)$, where n is the number of processors.[2] We prove the following theorem.

Theorem 1.1. *Given an arbitrary scheduling distribution $\mathbf{p} = (p_1, p_2, \ldots, p_n)$, and fixed operation length k, algorithms in* SCU *have system latency $\Theta(1/\|\mathbf{p}\|_2)$. Each process i has individual latency $\Theta(\|\mathbf{p}\|_2/p_i^2)$.*

Given the relatively complex structure of the algorithms, we find it surprising that such a simple characterization exists for general $\mathbf{p}$.

Analysis Overview. The argument behind these bounds is quite non-trivial, and can be summarized as follows. We express the balls-into-bins game corresponding to an algorithm, as a Markov chain, where each state is a possible configuration of occupancies of the bins, and each transition corresponds to a new ball being placed. An interesting feature of this chain is that a win by bin i also affects the state of bins containing k balls before i's win, which induces non-trivial correlations between occupancies of the bins. It would be enough to characterize the stationary distribution of this chain: the rate at which bin i wins is the stationary probability of bin i transitioning from k balls to 1 ball, while the win rate of the system is the sum of individual rates. These win rates are the reciprocals of individual and system latencies, respectively.

However, computing the stationary distribution of such a process for general $\mathbf{p}$, n, and k is infeasible, so we adopt a more elaborate strategy. We begin by bounding the system latency. Our main gadget for this part of the argument is to analyze a simplified chain, where all bin loads are considered modulo k. This choice is somewhat unintuitive, since this simplification obscures exactly the probability that a bin has k balls, which directly determines the bin winning rates. However, the stationary distribution of this simpler chain is easier to compute, and we show bounding it implies a clean $\sqrt{k}/\|\mathbf{p}\|_2$ lower bound on the expected system latency.

The system latency upper bound is more involved, and proceeds as follows. We split the execution into phases, where a phase is the interval between two wins. (The expected length of this interval is the system latency.) We consider the state of the chain at a *random* time t, and wish to bound the *residual* (remaining) phase length from such a time. We partition bins according to the number of balls they contain at t, where a bin is at level $1 \le \ell \le k+1$ if it still needs ℓ balls to reach $k+1$ and win the phase. Ideally, we would like to argue that there are many bins at low levels, and that these bins will dominate the phase competition, winning within a small number of steps.

However, since the distribution $\mathbf{p}$ may be unbalanced, the probability weight contained by each level ℓ at time t might not be well-concentrated for some ℓ. Instead, we take a more general approach, iterating Hoeffding bounds for each level ℓ, and merging them to show that the expected residual phase length has to be $O(1/\|\mathbf{p}\|_2)$, irrespective of the structure of $\mathbf{p}$. This is the most delicate part of the proof, which requires a careful analysis of the higher norms of $\|\mathbf{p}\|$ which turn out to be intimately related to the concentration of the occupancy distribution as well as the concentration

[2] Expressing complexity in terms of the number of participants n is standard for concurrent algorithms. For brevity, we write $\mathbf{p}$ instead of $\mathbf{p}(n)$.

of the system latency. We complete the argument by showing that, in expectation, the full phase length is bounded by twice the residual phase length from a random time t. The high probability bound is obtained similarly: specifically, we prove that system latency is $O(\text{ polylog}\, n/\|\mathbf{p}\|_2)$, with high probability.

We then turn our attention to the individual latencies, and in particular focus on upper and lower bounds on the probability w_i that a specific bin i wins a *random* phase. The lower bound argument uses a natural strategy: the simplified chain gadget suggests that, with constant probability, each bin i should start the phase at level two (needing just two more balls to win). We then get that w_i is $\Omega(p_i^2/\|\mathbf{p}\|_2^2)$, by assuming that all other balls also start at low levels.

We would like to make a similar argument to upper bound the probability that bin i wins a phase. The problem, however, is that, in contrast to the lower bound, where we essentially only need to understand the occupancy distribution of bin i at the beginning of a random phase, we now need to understand the occupancies of *all* bins to make sure that bin i gets enough competition. These occupancies may be quite complex as well as not well-concentrated, since we do not restrict $\mathbf{p}$ and k. We circumvent this problem by first considering the occupancy distribution at a random time step, which we partially understand from the system latency argument. Then we build a connection between the probability that the bin occupancies are unbalanced at *a random time* and the probability that the bin occupancies are unbalanced at *the beginning of a random phase*. The proof of this relation employs a complex adversarial counting argument, which may be of independent interest. This allows us to show that we can expect bin i to get enough competition on average, which implies a matching upper bound of $O(p_i^2/\|\mathbf{p}\|_2^2)$ on i's winning probability. Individual latency bounds follow from the winning probability bounds and the system latency bound.

Applications. Our result allows to relate the scheduling distribution $\mathbf{p}$ with the average-case performance of a lock-free algorithm, and provides simple answers to the questions on the relative latency advantage of processors (ratio of scheduling probabilities squared), and the worst-case scheduling distribution (uniform).

It has two additional practical applications: given the simple relation between $\mathbf{p}$ and the performance metrics, the question of determining the scheduling probabilities which minimize system latency, i.e., the optimal backoff scheme, under a set of individual latency constraints, becomes a well-defined optimization problem.

Second, a useful by-product of our technical framework is that we can understand the average-case behavior of various other concurrency patterns under arbitrary scheduling distributions. A particularly interesting pattern is that when, on a win by bin i, *all* bins get reset to 1, irrespective of their current bin count, which corresponds to the general family of *obstruction-free* concurrent algorithms, e.g. [3, 13], and to database transactions with optimistic concurrency control, e.g., [22, Chapter 7]. Our framework yields that the system latency of such algorithms is $\Theta(1/\|\mathbf{p}\|_k)$, and that the individual latency is This implies a separation in terms of average complexity between this algorithmic class and lock-free algorithms, for virtually any scheduling distribution $\mathbf{p}$.

Related Work. The first work to consider probabilistic schedulers in shared-memory was that of Aspnes [2], who showed that it is possible to solve consensus efficiently under a probabilistic scheduler model different from the one we consider. Alistarh, Censor-Hillel, and Shavit [1] introduced the *stochastic* scheduler model, and showed that it is sufficient to transform any bounded lock-free[3] algorithm into a wait-free one, with probability 1. They also isolated the *SCU* algorithmic pattern, and bounded its average-case performance for a *uniform* stochastic scheduler.

This implies bounds on the system and individual latencies for *SCU* given $\mathbf{p} = (1/n, 1/n, \ldots, 1/n)$. Specifically, they showed that system latency is $O(\sqrt{n})$ steps, while individual latency is $O(n\sqrt{n})$ steps, meaning that, on average, each processor will have to retry its scan-and-validate pattern $\Theta(\sqrt{n})$ times before succeeding. In brief, their analysis works by expressing the algorithm as a Markov chain, and noticing that, since both the operation code and the scheduling distribution are *symmetric*, the general Markov chain can be collapsed onto a simpler chain, called the *global* chain. In this chain, symmetric states corresponding to the same occupancy distribution (how many bins are at each occupancy level) are mapped onto the same global state. This mapping is a proper *lifting* [6, 11] of the general chain, and characterizing the stationary distribution of the global chain is enough to bound the latency of the algorithm. One main technical step is bounding the stationary probabilities of the global chain; this works by carefully characterizing the chain for $k = 2$, and then generalizing to general constants k by symmetry.

By contrast, we analyze algorithms in *SCU* under arbitrary distributions $\mathbf{p}$, which requires us to take a different approach. The first key difference is that we cannot employ Markov chain lifting, since states that are symmetric in terms of bin occupancies may now have completely different probabilities in the stationary distribution of the chain, as

[3] An algorithm is *bounded* lock-free if there exists a *finite* bound on the number of steps the system may schedule without completing an operation.

the bins have different probabilities of being selected. Similarly, we can no longer reduce the system latency analysis to the case $k = 2$, since the generalization to arbitrary k relies on symmetry of the distribution. Finally, in the symmetric case, system latency immediately determines the individual latencies, as two bins have exactly the same probability of winning an arbitrary phase. This is clearly no longer the case for non-uniform $\mathbf{p}$, and, in fact, determining the individual winning probabilities given the system latency is one of the main technical components of our argument. Our result implies the same results for uniform $\mathbf{p}$, although their lifting argument yields tighter constants in this case.

Multi-core processor architectures induce non-uniformity in the scheduling distribution, as memory access times for different threads are identical [16]. Recently, Dice et al. [7] exhibited scheduling non-uniformity for small-scale multi-threaded machines, due to caching effects. The impact of scheduling on the performance of concurrent algorithms is an active research topic, e.g., [8, 20].

Balls-into-bins processes [4] are popular abstractions for resource allocation problems, where balls (tasks, requests) are randomly allocated to bins (resources, processors). Such processes are relatively well understood in the uniform setting [4, 19], but recently there has been a growing interest in non-uniform settings, where either balls have different weights [24] or bins have correlated probabilities to be picked [10]. In contrast to our framework, however, one typically is concerned with bounding the maximum load, and employs the so-called two choice paradigm which involves picking *two* random bins and allocating the ball in the lesser loaded of the two.

2 Preliminaries

2.1 Modeling

The Shared Memory Model. In this model, n processes (threads) communicate through registers, on which they perform atomic operations, such as read, write, or compare-and-swap(CAS). A CAS operation takes three arguments $(R, \mathit{expectedVal}, \mathit{newVal})$, where R is the register on which it is applied, $\mathit{expectedVal}$ is the expected value of the register, and newVal is the new value to be written. The operation compares expVal with the current value v of R, and atomically updates R to newVal if the expected value matches the current value, i.e. $\mathit{expVal} = v$. In this case, we say that the CAS *succeeds*. Otherwise, the value of R is not updated, and the CAS *fails*.

Each algorithm implements a shared object, which is an abstraction providing a set of methods, each given by its sequential specification. The algorithm is composed of shared-memory steps and local computation. The order in which these steps are performed is controlled by a *scheduler*.

Stochastic Schedulers. In general, a scheduler for n processes can be defined by a triple (Π_t, A_t, θ). For each time step $t \geq 1$, Π_t is a probability distribution for scheduling the n processes at time t, and A_t is the subset of *possibly active* processes at time step t. At time step $t \geq 1$, the distribution Π_t gives, for every $i \in \{1, \ldots, n\}$ a probability p_i^t, with which process i is scheduled. The distribution Π_t may depend on arbitrary outside factors, such as the current state of the algorithm being scheduled. For every $t \geq 1$, the parameters ensure the following: (i) $\sum_{i=1}^{n} p_i^t = 1$; (ii) For every process $i \in A_t$, $p_i^t \geq \theta$; (iii) For every process $i \notin A_t$, $p_i^t = 0$; (iv) $A_{t+1} \subseteq A_t$. A scheduler (Π_t, A_t, θ) is *stochastic* if $\theta > 0$. In this paper, we consider a static *non-uniform* scheduler, which picks the next process to schedule using a fixed distribution $\mathbf{p} = (p_1, p_2, \ldots, p_n)$ at every time t.

Executions and Schedules. An execution is a sequence of operations performed by the processes. Since process steps are atomic, we can assume discrete time, where at every time unit a single process is scheduled. In a time unit, a process performs any number of local computations or coin flips, after which it issues a *step*, which consists of a single shared memory operation. Whenever a process is scheduled to take a new step by the scheduler, it performs its local computation and then executes its next step, as dictated by the algorithm. Thus, the *schedule* is a possibly infinite sequence of process identifiers. If process i is in position t in the sequence, then it is active at time t.

Progress Guarantees. An implementation is *lock-free (non-blocking)* if, for each time t in the execution, there exists a finite bound T_t such that *some* operation returns within the next T_t steps. An implementation is *wait-free* if, for every operation, there exists a finite bound T on the number of steps that the operation can take before returning.

2.2 Algorithmic Pattern and Complexity Measures

We now describe the *Single-CAS Universal* algorithmic pattern which is the focus of our analysis. This pattern can be used to provide a lock-free concurrent implementation of any sequential object, via a straightforward extension

```
1 Shared: registers R_1, R_2, ..., R_k
2   procedure operation()
3   while true do
4       v_1 ← R_1.read(); v_2 ← R_2.read(); ...; v_{k−1} ← R_{k−1}.read()
5       v_k ← R_k.read() /* Scan region */
6       v' ← new proposed state based on v_1, v_2, ..., v_k
7       flag ← CAS(R_k, v_k, v') /* Validation step */
8       if flag = true then
9           output success
```

Algorithm 1: The structure of the lock-free algorithms in SCU.

of Herlihy's universal construction [12]. This pattern is also the basis for several efficient implementations of shared objects, such as counters, lists, stacks, queues, or skip-lists, e.g. [9, 14, 25].

An operation using the SCU pattern consists of a *scan-and-validate* loop consisting of k steps reading shared registers $R_1, R_2, \ldots, R_k$, ending with a *validation* step which performs a *compare-and-swap* operation on the last read location R_k. (Please see Algorithm 1 for an illustration.) If this compare-and-swap operation succeeds, then the operation returns successfully. Otherwise, if the operation fails, and the process re-issues the operation.

Complexity Metrics. We address the following question: what is the performance of an arbitrary algorithm A in SCU, under an arbitrary non-uniform stochastic scheduler given by the distribution $\mathbf{p} = (p_1, p_2, \ldots, p_n)$?

We consider two natural performance measures. The first is *system latency*, i.e., the expected number of time steps between two completed operations by the system. The second is *individual latency*, i.e., the expected number of time steps between two committed operations by a *specific* process i. Individual latency is the correspondent of the *step complexity* metric, which measures the number of individual steps taken by an operation in order to complete.

2.3 Markov Chain Formulation and Preliminary Results

In the following, we fix an algorithm A in SCU. Although the algorithm may be implementing an arbitrary shared object, its structure is fixed, and follows the pattern in Algorithm 1. We assume that each process executes an infinite number of operations, and that each process executes its next operation as soon as it completes the current one.

Modeling the Algorithm. The first step is to notice that we can reduce A to the following balls-into-bins game. Consider a system of n bins, representing the processes, each starting with one ball. This state corresponds to the process being about to read the value of R_1. In each step, the scheduler allocates a new ball (i.e., a step) into one of the n bins: bin i gets the ball with probability p_i. Bins continue to receive balls until reaching k balls, which naturally corresponds to being about to perform the CAS operation in line 7 in Algorithm 1.

Following the algorithm, whenever the scheduler places a ball into a bin that already contains k balls, a new operation gets completed and we therefore record a *win* by the system and by the corresponding process. Crucially, on a win by process i, the bin corresponding to the process is reset to having one ball, whereas all the other bins which had exactly k balls before i's win will be reset to 0 balls.[4] All other bins maintain their ball count.

Markov Chain Formulation. It is easy to see that the system can be modeled as a discrete-time Markov chain. At the arrival of the t-th ball, a bin $I(t)$ is selected at random by drawing a sample from the distribution $\mathbf{p} = (p_1, p_2, \ldots, p_n)$. The state of the system is defined by the bin occupancies $X(t) = (X_1(t), X_2(t), \ldots, X_n(t))$ before the arrival of the t-th ball. If the selected bin has less than k balls, i.e., $X_{I(t)}(t) \in \{0, \ldots, k-1\}$, then we simply increment $X_{I(t)}$ by 1 and continue. Otherwise, if $X_{I(t)}(t) = k$, we reset $I(t)$ to one ball, i.e., $X_{I(t)}(t+1) = 1$. Further, we reset all other bins $j \neq I(t)$ with $X_j(t) = k$ so that $X_j(t+1) = 0$. The process of bin occupancies $\{X(t)\}_{t \geq 0}$ is a discrete-time Markov chain with the finite state space $\{0, 1, \ldots, k\}^n$.

Stationary Distribution and Ergodicity. Under the assumption that $p_i > 0$ for each bin i, the transition matrix of the Markov chain $\{X(t)\}_{t \geq 0}$ is irreducible, and thus, has a unique stationary distribution π (see, e.g., [5]).

The chain $\{X(t)\}_{t \geq 0}$ is not aperiodic. To resolve this issue, we simply consider a perturbed version that is parameterized with $\delta \in (0, 1)$ such that at each time step a new ball arrives with probability $1 - \delta$ and is assigned

[4]The reader may be wondering about the exact meaning of the 0 balls state. Notice that processes which had just read R, and were not scheduled to perform the CAS operation, will *fail* their CAS operation before being able to restart the loop.

to one of the bins according to the given distribution $\mathbf{p}$, and otherwise no ball arrives in the given time step and the occupancies of the bins remain unchanged. Such a perturbation results in a "lazy" Markov chain $\{X^\delta(t)\}_{t\geq 0}$ that has the same stationary distribution as the original chain, but unlike the original chain, it is aperiodic, and hence, ergodic. The increase of the system and individual expected latencies is only for a factor $1/(1-\delta)$, which can be made arbitrarily close to 1 by taking δ small enough. In what follows, we will tacitly assume that the given vector $\mathbf{p}$ is already chosen so that the chain $\{X(t)\}_{t\geq 0}$ is ergodic, i.e., irreducible and aperiodic.

Reset Points, Phases, and Latencies. For the Markov chain $\{X(t)\}_{t\geq 0}$, we say that a time-step t is a *reset* or a *win* point if at time-step t bin i is selected such that $X_i(t-1) = k$. The reset points correspond to a *point process* $T_0 \leq 0 < T_1 < T_2 < \ldots$ where T_r is the time-step of the r-th reset point.

A *phase* corresponds to the interval of time-steps starting at a reset point and ending in the time-step just before the next reset point. The phases are enumerated such that the 0-th phase corresponds to the interval of time-steps $[T_0, T_1)$, first phase corresponds to the interval of time-steps $[T_1, T_2)$ and so on. The length of the r-th phase is denoted by S_r. Since we are interested in the long-run average performance, we shall consider expected values with respect to the stationary distribution of the bin occupancies. We refer to a *random time* t to be an arbitrary time t under assumption that the initial bin occupancies at time 0 are according to the stationary distribution. Under this assumption, the distribution of the bin occupancies at any given time $t \geq 0$ is the stationary distribution.

We shall also consider the state of the bin occupancies embedded at the reset points. We refer to a *random phase* r to be an arbitrary phase r under assumption that the initial bin occupancies at time 0 are according to the stationary distribution of the bin occupancies at a reset point. We are interested in characterizing system latency τ, defined as the expected number of time-steps between two consecutive reset points, i.e., $\tau = \mathbf{E}[S_r]$. By the Palm inversion formula (see Lemma D.1)[5], the rate of reset points λ satisfies $\lambda = 1/\tau$. The individual latency τ_i for bin i is the expected number of time steps between two consecutive reset points in which bin i wins. Again, by the Palm inversion formula, the rate of wins of bin i, λ_i, satisfies $\lambda_i = 1/\tau_i$. The rates of reset points and individual reset points satisfy $\lambda = \sum_{i=1}^n \lambda_i$. The winning probability of bin i in a random phase r is denoted with w_i. By the Palm inversion formula, $\lambda_i = \lambda w_i$, for every bin i. The rate of reset points for a bin i satisfies $\lambda_i = p_i \mathbf{Pr}[X_i(t) = k]$, where t is a random time step.

Helper Results. We now state a couple of results which are used throughout the analysis. Their proofs are deferred to the Appendix. The first result characterizes the stationary distribution of a simplified stochastic process.

Lemma 2.1. *Let $Y(t) = (X_1(t) \bmod k, X_2(t) \bmod k, \ldots, X_n(t) \bmod k)$. Then, $Y(t)$ is an ergodic, discrete-time Markov chain on the state space $\{0, 1, \ldots, k-1\}^n$ with uniform stationary distribution π, i.e.*

$$\lim_{t\to\infty} \mathbf{Pr}[Y(t) = \mathbf{x} \mid Y(0) = \mathbf{y}] = \pi(\mathbf{x}) = 1/k^n, \quad \textit{for any } \mathbf{x}, \mathbf{y} \in \{0, 1, \ldots, k-1\}^n.$$

The second result characterizes the system latency and probability of winning a random phase for a simpler process, where all participants get reset to 1 on a win. The analysis of this process is non-trivial, and may be of independent interest, as it is related to non-uniform birthday processes, e.g. [21]. Its proof can be found in Appendix C.

Theorem 2.2. *The length S_r of a random phase r and the probabilities w_i that bin i wins an arbitrary phase, in the process where all bins reset to one ball on a win, satisfy:*

$$\mathbf{E}[S_r] \in \left[\frac{(k!/2)^{1/k}}{2\|\mathbf{p}\|_k}, \frac{8ek^k}{\|\mathbf{p}\|_k}\right] \textit{ and } w_i = \Theta\left(\frac{p_i^k}{\|\mathbf{p}\|_k^k}\right) \textit{ for every bin } 1 \leq i \leq n.$$

3 Analysis of System Latency

Proof Strategy. We are interested in upper and lower bounds on system latency τ. Determining these quantities from the stationary distribution of the Markov chain described in the previous section would be sufficient. Unfortunately, we will not be able to determine such bounds directly, as the occupancies of individual bins are highly correlated.

[5] A well-known formula in the context of renewal and regenerative processes that relates expectations of a stochastic process with respect to the stationary distribution of the state at an arbitrary time and the stationary distribution of the state at a special point in time, see e.g. [5].

Instead, we take a more roundabout approach: we first consider the stationary distribution of the simplified random process described in Lemma 2.1, in which bin occupancies are taken modulo k. Analyzing such a process may appear useless at first, as it obscures the probability that a bin contains k balls, which is essential for determining λ_i.

However, a first application of this characterization is a clean $\sqrt{k}/\|\mathbf{p}\|_2$ lower bound on system latency τ for the full stochastic process. The proof can be found in the Appendix, and follows by writing out the balance equations for state $X_i = 0$, combining with Lemma 2.1, and a simple convexity argument.

Theorem 3.1. *The expected system latency satisfies* $\tau \geq \sqrt{k}/\|\mathbf{p}\|_2$.

The proof of the upper bound is more complex, and works as follows. We proceed in two steps. The first step is to assume a random time t, and to bound the *residual phase length* $S(t)$, i.e., the remaining number of steps until an operation completes, ending the phase. The second step is to bound the length of a phase depending on this quantity. (Note that bounding the residual phase length gives a *stronger* measure of performance than the expected phase length.)

To bound the residual phase length, our starting tool is Lemma 2.1, which provides a handle on the occupancy distribution modulo k at time t. However, the main difficulty with bounding $S(t)$ is that the occupancy distribution at time t may be quite irregular depending on the "shape" of the vector $\mathbf{p}$. For instance, we may expect bins which already contain several balls at time t to complete an operation first, since they have an advantage, but they may happen to have a low total probability. Our argument will take into account these possibilities, iterating Hoeffding bounds for each occupancy level. We show that, irrespective of irregularities, the residual phase length is always $O(1/\|\mathbf{p}\|_2)$.

Lemma 3.2. *For a random time t, the residual phase length $S(t)$ satisfies $\mathbf{E}[S(t)] = O(1/\|\mathbf{p}\|_2)$. Moreover, $\mathbf{Pr}[S(t) \leq C \cdot (\log n)^{1+\frac{\log k}{2}}/\|\mathbf{p}\|_2] \geq 1 - 1/n$, for a constant $C > 0$.*

Proof. We partition the bins into *levels* $1, 2, \ldots, k+1$, where a bin is at level ℓ at time t if it needs to acquire ℓ balls in order to win, i.e., it contains $k+1-\ell$ balls at t. Since we are interested in upper bounding the phase length, we can assume that there are no bins at level 1 at time t, as those could only shorten the phase.

For each bin i and level $2 \leq \ell \leq k+1$, we define the indicator random variable $Z_{\ell,i}(t)$ to equal p_i^ℓ if bin i is at level ℓ at time t, and 0, otherwise. Further, define $Z_\ell(t) = \sum_{i=1}^n Z_{\ell,i}(t)$. By Lemma 2.1, we know that $\mathbf{E}[Z_\ell(t)] = \frac{1}{k}\|\mathbf{p}\|_\ell^\ell$ and that $Z_{\ell,1}(t), Z_{\ell,2}(t), \ldots, Z_{\ell,n}(t)$ are independent events for every $2 \leq \ell \leq k+1$. Hence, for fixed ℓ, we apply Hoeffding's inequality to variables $(Z_{\ell,i}(t))_i$ to obtain the following bound

$$\mathbf{Pr}\left[Z_\ell(t) \leq \|\mathbf{p}\|_\ell^\ell/(2k)\right] \leq \exp\left(-\frac{1}{2k^2}\frac{(\sum_{i=1}^n p_i^\ell)^2}{\sum_{i=1}^n p_i^{2\ell}}\right) = \exp\left(-\frac{1}{2k^2}\frac{\|\mathbf{p}\|_\ell^{2\ell}}{\|\mathbf{p}\|_{2\ell}^{2\ell}}\right). \quad (1)$$

Recall that the phase ends as soon as, for some level ℓ, a bin that is at level ℓ at time t acquires ℓ balls. By Theorem C.2, we have that, conditional on the values of $(Z_{\ell,i}(t))_{\ell,i}$ the expected residual phase length is

$$\mathbf{E}[S(t) \mid (Z_{\ell,i}(t))_{\ell,i}] = O\left(\min_{2\leq\ell\leq k+1}\frac{1}{Z_\ell(t)^{1/\ell}}\right). \quad (2)$$

We now focus on bounding the left-hand side quantity in terms of $1/\|\mathbf{p}\|_2$.

For $2 \leq \ell \leq k$, let $\mathcal{E}_\ell$ be the event that $Z_\ell(t) \leq \|\mathbf{p}\|_\ell^\ell/(2k)$. Let j be the index of the first event $\mathcal{E}_\ell$ that holds when we consider the events $\mathcal{E}_2, \mathcal{E}_3, \ldots, \mathcal{E}_k$ in increasing order of parameter ℓ if any exists, and let j be equal to $k+1$, otherwise. In other words, the norm bounds for all levels $< j$ have failed, and the bound holds at j. Combining the norm bound with Equation (2), it follows that $\mathbf{E}[S(t) \mid \overline{\mathcal{E}_2} \wedge \overline{\mathcal{E}_3} \wedge \ldots \wedge \overline{\mathcal{E}_{j-1}} \wedge \mathcal{E}_j] = O(1/\|\mathbf{p}\|_j)$.

Applying the law of total expectation, it follows that there exists a constant $C > 0$ such that

$$\mathbf{E}[S(t)] \leq C\left(\frac{1}{\|\mathbf{p}\|_2} + \sum_{\ell=2}^k \mathbf{Pr}\left[\overline{\mathcal{E}_2} \wedge \overline{\mathcal{E}_3} \wedge \ldots \wedge \overline{\mathcal{E}_\ell}\right]\frac{1}{\|\mathbf{p}\|_{\ell+1}}\right). \quad (3)$$

Bounding the probability term for fixed ℓ and applying the bound (1) we obtain

$$\mathbf{Pr}\left[\overline{\mathcal{E}_2} \wedge \overline{\mathcal{E}_3} \wedge \ldots \wedge \overline{\mathcal{E}_\ell}\right] \leq \min_{j=2}^{\ell}\left\{\mathbf{Pr}[\overline{\mathcal{E}_j}]\right\} \leq \left(\prod_{j=2}^{\ell}\mathbf{Pr}[\overline{\mathcal{E}_j}]\right)^{1/(\ell-1)} \leq \prod_{j=2}^{\ell}\exp\left(-\frac{1}{2k^2(\ell-1)}\frac{\|\mathbf{p}\|_j^{2j}}{\|\mathbf{p}\|_{2j}^{2j}}\right).$$

Bounding the last term using the fact that $\exp(-x) \le 1/x$ for $x > 0$, and $\|\mathbf{p}\|_j/\|\mathbf{p}\|_{j+1} \ge 1$, we have

$$\prod_{j=2}^{\ell} \exp\left(-\frac{1}{2k^2(\ell-1)} \frac{\|\mathbf{p}\|_j^{2j}}{\|\mathbf{p}\|_{2j}^{2j}}\right) \quad \le \prod_{j=2}^{\ell} 2(\ell-1)k^2 \tfrac{\|\mathbf{p}\|_{j+1}}{\|\mathbf{p}\|_j} \le (2(\ell-1)k^2)^{\ell-1} \tfrac{\|\mathbf{p}\|_{\ell+1}}{\|\mathbf{p}\|_2}.$$

Combining with Equation (3), it follows that there exists a constant $C' > 0$ such that

$$\mathbf{E}[S(t)] \le C' \left(\frac{1}{\|\mathbf{p}\|_2} + \sum_{\ell=2}^{k} \frac{\|\mathbf{p}\|_{\ell+1}}{\|\mathbf{p}\|_2} \cdot \frac{1}{\|\mathbf{p}\|_{\ell+1}}\right) \le C' \cdot \frac{k}{\|\mathbf{p}\|_2} = O\left(\frac{1}{\|\mathbf{p}\|_2}\right).$$

The proof of the high probability bound is similar, and can be found in the appendix. □

Final Step. To obtain the upper bounds on the phase length, we relate the residual phase length $S(t)$ at a random time t to the expected phase length S_r of a random phase r. We have that $\mathbf{E}[S_r] \le 2\,\mathbf{E}[S(t)]$, and $\mathbf{Pr}[S_r > 2\theta] \le 2\,\mathbf{Pr}[S(t) > \theta]$, for $\theta \ge 0$ (Lemma A.1). This completes the proof of the phase upper bound.

Theorem 3.3. *The expected system latency satisfies* $\tau = O(1/\|\mathbf{p}\|_2)$.

4 Analysis of Individual Latencies

4.1 Upper Bound for Winning Probabilities

We prove the following bound on the probability that bin i wins a phase, which will imply an individual latency bound of $\Omega(\|\mathbf{p}\|_2/p_i^2)$ for bin i.

Theorem 4.1. *For every bin* i, *the winning probability satisfies* $w_i = O(p_i^2/\|\mathbf{p}\|_2^2)$.

Proof Outline. The proof of this result is quite involved, so we first provide an outline for the case $k = 2$. Fix an arbitrary bin i. Let W_r be the event that in a random phase r bin i wins, so $w_i = \mathbf{Pr}[W_r]$. A time step t is defined to be bad if $\sum_{i=1}^{n} p_i^2 \mathbf{1}_{Y_i(t)=1} \le \|\mathbf{p}\|_2^2/16$. Similarly, a phase r is defined to be bad if its first time step is bad. Intuitively, if a phase is not bad, then there is "enough competition" which limits the winning probability of i. Let B_r denote the event that phase r is bad. By a simple conditioning, we have $\mathbf{Pr}[W_r] = \mathbf{Pr}[\overline{B}_r] \cdot \mathbf{Pr}[W_r \mid \overline{B}_r] + \mathbf{Pr}[B_r] \cdot \mathbf{Pr}[W_r \mid B_r]$. We shall establish the following three claims:

$$\mathbf{Pr}[B_r] \le \exp\left(-C \frac{\|\mathbf{p}\|_2^2}{\|\mathbf{p}\|_3^2}\right), \text{ for a constant } C > 0, \tag{4}$$

$$\mathbf{Pr}[W_r \mid \overline{B}_r] = O\left(\frac{p_i^2}{\|\mathbf{p}\|_2^2}\right) \qquad \text{and} \qquad \mathbf{Pr}[W_r \mid B_r] = O\left(\frac{p_i^2}{\|\mathbf{p}\|_3^2}\right). \tag{5}$$

The asserted upper bound follows then by combining these bounds and the fact $\exp(-x) \le 1/x$ for $x > 0$,

$$\mathbf{Pr}[W_r] \le \exp\left(-C \frac{\|\mathbf{p}\|_2^2}{\|\mathbf{p}\|_3^2}\right) \cdot \frac{p_i^2}{\|\mathbf{p}\|_3^2} + \frac{p_i^2}{\|\mathbf{p}\|_2^2} = O\left(\frac{p_i^2}{\|\mathbf{p}\|_2^2}\right).$$

The inequality in (4) is obtained by showing first that the probability that a random time t is bad is at most $\exp(-C'\|\mathbf{p}\|_2/\|\mathbf{p}\|_3)$ for a constant $C' > 0$, and then showing that the probability that a random phase r is bad is upper bounded by a similar bound. The first upper bound in (5) is obtained by conservatively assuming that at the beginning of phase r, bin i has one ball and exploiting that the probability mass on the bins having one ball is large enough, and then applying Theorem C.3. The second upper bound in (5) is obtained by assuming that bin i has one ball and all other bins have zero balls, and applying Corollary C.4.

For the case $k \ge 3$, we need a more careful analysis, as the (conditional) winning probabilities could be anything between the values $p_i^2/\|\mathbf{p}\|_j^2$ for $2 \le j \le k+1$. Corresponding to these values, we define different gradients of "badness", and derive bounds as in (4) and (5) for each gradient separately. Then all these bounds are charged against each other and the proof is completed in a similar way as in Lemma 3.2. We give the full proof of Theorem 4.1 in the appendix and focus here on establishing the key step, Eq. (4), and its version for $k \geq 3$.

Probability of a Bad Phase. Our goal is to show that the probability that a random phase is bad is small. In order to capture the general case $k \geq 3$, we first have to formalize the notion of ℓ-badness of a time step and introduce some further pieces of notation. Similar to the proof of Lemma 3.2, we partition bins into levels $1, 2, \ldots, k+1$, where bins at level ℓ need to acquire ℓ balls in order to win. For each bin i and level $2 \leq \ell \leq k+1$, we define the indicator random variable $Z_{\ell,i}(t)$ to equal p_i^ℓ if bin i is at level ℓ at time t, and 0, otherwise. Further, define $Z_\ell(t) = \sum_{i=1}^n Z_{\ell,i}(t)$.

Definition 4.2. *For every level $2 \leq \ell \leq k$, a time step t is*

1. ℓ-bad *if $Z_\ell(t) < \|\mathbf{p}\|_\ell^\ell/(8k)$,*
2. ℓ-good *if $Z_\ell(t) \geq \|\mathbf{p}\|_\ell^\ell/(8k)$,*
3. ℓ-perfect *if all time steps in $[t, t + 1/(32k\|\mathbf{p}\|_2)]$ are ℓ-good.*

Similarly as before, a phase is ℓ-bad, ℓ-good or ℓ-perfect if the first time step of that phase is ℓ-bad, ℓ-good or ℓ-perfect. The key point behind the definition of ℓ-perfect that it enforces a particular structure on the time steps; every block of ℓ-perfect time steps is followed up with exactly $1/(32k\|\mathbf{p}\|_2)$ rounds which are ℓ-good (but not ℓ-perfect), after which there is at least one ℓ-bad round. The following lemma is the key ingredient of the proof.

Lemma 4.3 (Key Lemma)**.** *For every level $2 \leq \ell \leq k$, a random phase is ℓ-bad w. p. at most* $\exp(-\Omega((\frac{\|\mathbf{p}\|_\ell}{\|\mathbf{p}\|_{\ell+1}})^4))$.

Proof. Let $B^\ell(T)$ be the number of ℓ-bad phases and let $N(T)$ be the number of phases in the time interval $[0, T)$. By the ergodicity of the Markov chain of bin occupancies and Theorem 3.3,

$$\mathbf{Pr}[\text{phase } r \text{ is } \ell\text{-bad}] = \lim_{T\to\infty} \frac{B^\ell(T)}{N(T)} \qquad \text{and} \qquad \lim_{T\to\infty} \frac{N(T)}{T} = \Omega(\|\mathbf{p}\|_2),$$

where the limits hold with probability 1. Let $L^\ell(T)$ be the number of non-perfect times steps in the time interval $[0, T)$. From Lemma 4.5 below it follows that $\lim_{T\to\infty} B^\ell(T)/L^\ell(T) = O(\|\mathbf{p}\|_2)$ with probability 1. Finally, it follows by Lemma 4.4 that

$$\lim_{T\to\infty} \frac{L^\ell(T)}{T} \leq \exp\Big(-\Omega\big((\frac{\|\mathbf{p}\|_\ell}{\|\mathbf{p}\|_{\ell+1}})^4\big)\Big)$$

with probability 1. Combining these relations completes the proof. ☐

Lemma 4.4. *For every level $2 \leq \ell \leq k$, a random time t is ℓ-perfect w. p. at least* $1 - \exp\Big(-\Omega\big((\frac{\|\mathbf{p}\|_\ell}{\|\mathbf{p}\|_{\ell+1}})^4\big)\Big)$.

The proof of Lemma 4.4 follows by first exploiting that for a random time t, the occupancy distribution modulo k is uniform, which allows us to apply Hoeffding's bound to lower bound $Z_\ell(t)$. Then we lower bound the set of bins at level ℓ at time step t which do not receive any ball until time $t + 1/(32k\|\mathbf{p}\|_2)$. Combining these two parts yields the statement of the lemma (see appendix for further details).

Lemma 4.5. *For every level $2 \leq \ell \leq k$ and a random time t,*

$$\mathbf{Pr}[\textit{ time step } t \textit{ is beginning of a } \ell\textit{-bad phase }] \leq O(\|\mathbf{p}\|_2) \cdot \mathbf{Pr}[\textit{ time step } t \textit{ is not } \ell\textit{-perfect }].$$

The proof of this lemma relies on a careful comparison between the point process associated to ℓ-bad phases, and the point process associated to ℓ-(non-)perfect time steps. Intuitively, we will show that, on average, every bad phase can be charged $\Omega(1/\|\mathbf{p}\|_2)$ non-perfect time steps, which then implies the statement of the lemma.

Proof of Lemma 4.5. In this proof we are going to relate ℓ-non-perfect time steps and ℓ-bad phases (since $\ell \in \{2, \ldots, k-1\}$ will be fixed throughout this part, we will often simply say non-perfect and bad). To this end we will consider a specific process which simulatenously counts the number of bad phases and lower bounds the number of non-perfect time steps. This process will use a (non-complete) partitioning of phases into set of phases, called epochs.

Definition 4.6. *An epoch is defined to be an interval of consecutive phases that starts at the beginning of a bad phase and ends at the end of the first occurance of a phase that contains at least one perfect time step.*

Starting from a fixed perfect time step (w.l.o.g. time 0), we can enumerate all epochs by $E_1, E_2, \ldots$. The j-th phase of the e-th epoch is denoted by $E_{e,j}$, whose length is $|E_{e,j}|$. Finally, $|E_e| \geq 1$ denotes the size of epoch e, i.e., the number of phases it contains. An illustration of these definitions can be found in Figure 1.

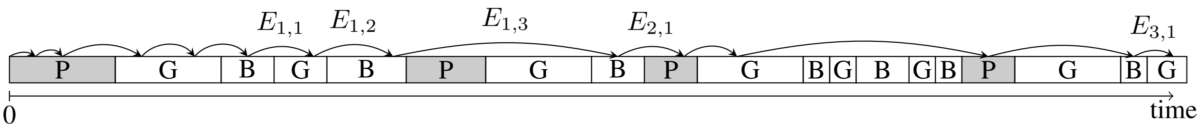

Figure 1: Illustration of phases viewed as a point process in time, which can be of three different types: P (perfect), G (good but not perfect) and B (bad). Arrows with a label $E_{e,j}$ form phase j in epoch e. Phases without a label do not belong to any epoch.

Counting the Ratio Phase-by-Phase. As suggested in Figure 1, there is a connection between the phase lengths and the number of non-perfect time steps. In particular, every phase of an epoch except for the last one contains only non-perfect time steps. To elaborate on this link, we next describe a process which follows the partitioning into epochs to update two counters. First, a counter B adds up all phases which are part of the e-th epoch. Simultaneously, a second counter L adds up all phase lengths $|E_{e,j}|$ except the last one as well as the $1/(32k\|\mathbf{p}\|_2)$ good but non-perfect time steps that prelude any new epoch. The precise description of these updates can be found in Process 2.

```
e = 0, j = 0 /* counters for epoch and phase */
L = 0, B = 0 /* counters for non-perfect rounds and bad phases */
while true do
    Jump to the next bad phase
    e = e + 1, j = 0
    repeat
        j = j + 1
        Allocate balls until phase j of epoch i ends
        if j = 1 then L = L + 1/(32k||p||_2), B = B + 1
        if j ≥ 2 then L = L + |E_{e,j-1}|, B = B + 1
    until phase j contained a perfect time step
```

Process 2: The original counting process and its phase-based updates.

The next lemma shows that the counters B and L are indeed related to the number of bad phases and the number of non-perfect rounds. It basically follows from the definition of an epoch (see appendix for details).

Lemma 4.7. *Consider Process 2 running until the f-th epoch is completed, and let $B = B(f)$ and $L = L(f)$ be the values of the counters B and L at this time. Then the following two statements hold. First, $B(f)$ counts exactly the total number of bad phases that may occur until the last time step of the f-th epoch. Second, the expression $L(f) = \sum_{e=1}^{f} \big(\frac{1}{32k\|\mathbf{p}\|_2} + \sum_{j=1}^{|E_e|-1} |E_{e,j}| \big)$ lower bounds the number of non-perfect time steps which belong to a phase of the first f epochs.*

Counting the Ratio Epoch-by-Epoch. The problem with analyzing Process 2 is that we do not have a good control on its actual length in terms of phases or time steps. In fact, we could think of facing an online adversary who is allowed to see the evolution of all phase lengths and then able to end the epoch after an arbitrary phase. However, even the power of this adversary would be limited, as every phase will take $\Omega(1/\|\mathbf{p}\|_2)$ with constant probability > 0, regardless of the occupancy distribution at the beginning of that phase. This motivates the definition of a modified process, where we let the epoch length be the maximum number x so that more than half of its phase lengths are less than some threshold $1/(16k\|\mathbf{p}\|_2)$. If $x > 0$, then we conservatively assume that the epoch runs for x phases all of which have length 0. If $x = 0$, then we assume that the epoch consists of one phase of length $1/(128k\|\mathbf{p}\|_2)$.

To define the modified process formally, we generate for the e-th epoch with phase j a separate (infinite) sequence of bin samples denoted by $M_{e,j} \in \{1, \ldots, n\}^{\mathbb{N}}$. Here, $M_{e,j}(t)$ is the bin where the t-th ball of phase j in epoch E_e is placed, in particular, for any bin i, we have $\mathbf{Pr}[M_{e,j}(t) = i] = p_i$. Further, for every pair e, j, we use an indicator variable $Z_{e,j}$ which is one iff the bin samples $M_{e,j}$ ensure that no bin gets two balls within $1/(16k\|\mathbf{p}\|_2)$ time steps.

$e = 0, \widetilde{L} = 0, \widetilde{B} = 0$ `/* counters for epoch, non-perfect rounds and bad phases */`
$M_{e,j} \in \{1, \ldots, n\}^{\mathbb{N}}$ `/* random sequence for bin samples in phase` j `of the` e`-th epoch */`
while true **do**
    $e = e + 1$
    $\widetilde{S}_e = \max\{x \geq 1 : \sum_{j=1}^{x} Z_{e,j} < k/2\}$ `/* defined by` $M_{e,j}$ `for` $j = 1, 2, \ldots$ `*/`
    if $\widetilde{S}_e = 0$ **then** $\widetilde{L} = \widetilde{L} + 1/(128k\|\mathbf{p}\|_2)$, $\widetilde{B} = \widetilde{B} + 1$
    if $\widetilde{S}_e \neq 0$ **then** $\widetilde{L} = \widetilde{L} + 0$, $\widetilde{B} = \widetilde{B} + \widetilde{S}_e$

Process 3: The modified process and its epoch-based updates.

We proceed to establish a coupling between the two processes which implies that L/B majorises $\widetilde{L}/\widetilde{B}$. A random variable X is stochastically smaller (is majorized by) Y, in symbols $X \preceq Y$, if $\mathbf{Pr}[X \geq \theta] \leq \mathbf{Pr}[Y \geq \theta]$ for every θ.

Lemma 4.8. *Consider the original process and modified process both running until epoch f is completed, and let $B(f)$, $L(f)$, $\widetilde{B}(f)$, $\widetilde{L}(f)$ be the value of the four counters at this time. Then $L(f)/B(f) \succeq \widetilde{L}(f)/\widetilde{B}(f)$.*

The proof of this lemma is based on a coupling where the original process uses the same set of bin samples $M_{e,j}$ as the modified process (see appendix for details). Finally, we analyze the distribution of the counters $\widetilde{B}$ and $\widetilde{L}$.

Lemma 4.9. *For the modified process (Process 3), the following statement holds.*

- *For any $\theta \geq 1$ and epoch e, $\mathbf{Pr}[\widetilde{S}_e \geq \theta] \leq (1/4)2^{-\theta}$, thus, in particular, $\mathbf{Pr}[\widetilde{S}_e < \infty] = 1$.*
- *Let $\widetilde{B} = \widetilde{B}(f)$ and $\widetilde{L} = \widetilde{L}(f)$ be the values of the counters $\widetilde{B}$ and $\widetilde{L}$ until epoch f is completed. Then, $\widetilde{L}(f) \succeq \mathsf{Bin}(f, 1/2)/(128k\|\mathbf{p}\|_2)$, and $\widetilde{B}(f) \preceq \mathsf{Geo}(1/2) \cdot f$.*

Completing the Proof of Lemma 4.5. First, we consider $\widetilde{L}(f)/\widetilde{B}(f)$ for a large value of epoch f. By Lemma 4.9, $\mathbf{E}[\widetilde{L}(f)] \geq (f/2)/(128k\|\mathbf{p}\|_2)$ and $\mathbf{E}[\widetilde{B}(f)] \leq 2f$. Hence, by the ergodicity of the Markov chain the following limits hold with probability 1,

$$\lim_{f\to\infty} \widetilde{L}(f) \geq f \cdot \frac{1}{192k\|\mathbf{p}\|_2} \text{ and } \lim_{f\to\infty} \widetilde{B}(f) \leq 3f, \text{ implying } \lim_{f\to\infty} \frac{L(f)}{B(f)} \geq \lim_{f\to\infty} \frac{\widetilde{L}(f)}{\widetilde{B}(f)} \geq \frac{1}{576k\|\mathbf{p}\|_2},$$

where the first inequality follows from Lemma 4.8. By Lemma 4.7, we know that $L(f)$ is a lower bound on the number of non-perfect time steps until the end of epoch f, and $B(f)$ counts the number of bad phases until the end of epoch f. Hence by considering the beginning of bad phases as a point process in time, it follows that

$$\mathbf{Pr}[\text{ time step } t \text{ is beginning of a bad phase }] \leq 576k \cdot \|\mathbf{p}\|_2 \cdot \mathbf{Pr}[\text{ time step } t \text{ is not perfect }].$$

□

Lower Bound on Winning Probabilities. We know that for a random time t the probability that bin i has $k-2$ balls is exactly $1/k$. We show that this condition continues to hold from time t until the end of the current phase with a constant probability. This will follow by considering the additional events that (i) bin i does not receive a ball in the time interval $[t, t+s]$ where $s = \Theta(1/\|\mathbf{p}\|_2)$ and (ii) the event that the next phase starts in that interval. By reasoning about the point process, we can deduce that for a constant fraction of phases, bin i has $k-2$ balls, and the claim then follows by Theorem C.3 which shows that bin i has a probability of $\Omega(p_i^2/\|\mathbf{p}\|_2^2)$ for winning such a phase.

Theorem 4.10. *For any bin i, the winning probability satisfies $w_i = \Omega(p_i^2/\|\mathbf{p}\|_2^2)$.*

5 Applications and Future Directions

Our characterization has two non-trivial applications. The first is given by the following practical question: given a set of lower bound constraints on the performance of individual threads (individual latency upper bounds), what is the scheduling distribution which minimizes *system* latency? Modifying the scheduling distribution is commonly done through back-offs. By Theorem 1.1, this now corresponds to the simple optimization problem of maximizing the 2-norm of the scheduling probabilities $\mathbf{p}$, subject to a set of lower bounds on the scheduling probabilities.

The second application comes by noticing that the simple process where all bins are reset to 1 on a win models *obstruction-free* concurrent algorithms, which guarantee progress only in the absence of contention, and may abort operations otherwise. Theorem C.2 implies that the system latency of an obstruction-free algorithm is $\Theta(1/\|\mathbf{p}\|_k)$, where k is the operation length. This provides a complexity separation between obstruction-free and lock-free algorithms, under stochastic schedulers, for most values of the scheduling distribution $\mathbf{p}$.

Summing up, we have given general framework for competitive non-uniform balls-into-bins processes. While we focused on a particular application, lock-free algorithms, we believe that our techniques can be used to model a broader class of algorithms and processes related to scheduling contended applications. Hence, a natural but challenging question would be to find a general characterization, which parameterizes the winning probabilities and phase lengths in terms of the reset policies.

References

[1] Dan Alistarh, Keren Censor-Hillel, and Nir Shavit. Are lock-free concurrent algorithms practically wait-free? In *Proceedings of the 46th ACM Symposium on Theory of Computing*, STOC'14, pages 714–723, 2014. Full version containing measurements available at: http://arxiv.org/abs/1311.3200.

[2] James Aspnes. Fast deterministic consensus in a noisy environment. *J. Algorithms*, 45(1):16–39, 2002.

[3] Hagit Attiya, Rachid Guerraoui, Danny Hendler, and Petr Kuznetsov. The complexity of obstruction-free implementations. *J. ACM*, 56(4):24:1–24:33, July 2009.

[4] Yossi Azar, Andrei Z. Broder, Anna R. Karlin, and Eli Upfal. Balanced allocations. *SIAM J. Comput.*, 29(1):180–200, 1999.

[5] Pierre Bremaud. *Markov Chains: Gibbs Fields, Monte Carlo Simulation, and Queues (Text in Applied Mathematics)*. Springer, 2nd edition, 2001.

[6] Fang Chen, László Lovász, and Igor Pak. Lifting markov chains to speed up mixing. In *Proceedings of the 31st Annual ACM Symposium on Theory of Computing*, STOC'99, pages 275–281.

[7] Dave Dice, Virendra J. Marathe, and Nir Shavit. Brief announcement: Persistent unfairness arising from cache residency imbalance. In *Proceedings of the 26th ACM Symposium on Parallelism in Algorithms and Architectures*, SPAA'14, pages 82–83, 2014.

[8] Faith Ellen, Panagiota Fatourou, Joanna Helga, and Eric Ruppert. The amortized complexity of non-blocking binary search trees. In *Proceedings of the 33rd ACM Symposium on Principles of Distributed Computing*, PODC'14, pages 332–340, 2014.

[9] Keir Fraser and Timothy L. Harris. Concurrent programming without locks. *ACM Trans. Comput. Syst.*, 25(2), 2007.

[10] Brighten Godfrey. Balls and bins with structure: balanced allocations on hypergraphs. In *Proceedings of the 19th ACM-SIAM Symposium on Discrete Algorithms*, SODA'08, pages 511–517, 2008.

[11] Thomas P. Hayes and Alistair Sinclair. Liftings of tree-structured markov chains. In *Proceedings of the 13th International Conference on Approximation, and 14th International Conference on Randomization, and Combinatorial Optimization: Algorithms and Techniques*, APPROX/RANDOM'10, pages 602–616, 2010.

[12] Maurice Herlihy. Wait-free synchronization. *ACM Transactions on Programming Languages and Systems*, 13(1):123–149, January 1991.

[13] Maurice Herlihy, Victor Luchangco, and Mark Moir. Obstruction-free synchronization: Double-ended queues as an example. In *Proceedings of the 23rd International Conference on Distributed Computing Systems*, ICDCS'03, pages 522–529, 2003.

[14] Maurice Herlihy and Nir Shavit. *The art of multiprocessor programming*. Morgan Kaufmann, 2008.

[15] Maurice Herlihy and Nir Shavit. On the nature of progress. In *Proceedings of the 15th International Conference on Principles of Distributed Systems*, OPODIS'11, pages 313–328, 2011.

[16] Christoph Lameter. Numa (non-uniform memory access): An overview. *Queue*, 11(7):40:40–40:51, July 2013.

[17] Nancy A. Lynch. *Distributed Algorithms*. Morgan Kaufmann, 1996.

[18] Maged M. Michael and Michael L. Scott. Simple, fast, and practical non-blocking and blocking concurrent queue algorithms. In *Proceedings of the 15th ACM Symposium on Principles of Distributed Computing*, PODC'96, pages 267–275, 1996.

[19] Michael Mitzenmacher and Eli Upfal. *Probability and Computing: Randomized Algorithms and Probabilistic Analysis*. Cambridge University Press, 2005.

[20] Shuai Mu, Yang Cui, Yang Zhang, Wyatt Lloyd, and Jinyang Li. Extracting more concurrency from distributed transactions. In *11th USENIX Symposium on Operating Systems Design and Implementation*, OSDI'14, 2014.

[21] Thomas S. Nunnikhoven. A birthday problem solution for nonuniform birth frequencies. *The American Statistician*, 46(4):270–274, 1992.

[22] Raghu Ramakrishnan and Johannes Gehrke. *Database Management Systems*. McGraw-Hill, Inc., New York, NY, USA, 3rd edition, 2003.

[23] Michael L. Scott. *Shared-Memory Synchronization*. Synthesis Lectures on Computer Architecture. Morgan & Claypool Publishers, 2013.

[24] Kunal Talwar and Udi Wieder. Balanced allocations: the weighted case. In *Proceedings of the 39th ACM Symposium on Theory of Computing*, STOC'07, pages 256–265, 2007.

[25] R. K. Treiber. Systems programming: Coping with parallelism. Technical Report RJ 5118, IBM Almaden Research Center, 1986.

A Omitted Details from Section 3

Lemma 2.1. *Let* $Y(t) = (X_1(t) \bmod k, X_2(t) \bmod k, \ldots, X_n(t) \bmod k)$*. Then,* $Y(t)$ *is an ergodic, discrete-time Markov chain on the state space* $\{0, 1, \ldots, k-1\}^n$ *with uniform stationary distribution* π*, i.e.*

$$\lim_{t\to\infty} \mathbf{Pr}[Y(t) = \mathbf{x} \mid Y(0) = \mathbf{y}] = \pi(\mathbf{x}) = 1/k^n, \quad \textit{for any } \mathbf{x}, \mathbf{y} \in \{0, 1, \ldots, k-1\}^n.$$

Proof. For each bin i and time t, we have

$$Y_i(t+1) = \begin{cases} Y_i(t) + 1 \bmod k & \text{with probability } p_i \\ Y_i(t) & \text{with probability } 1 - p_i \end{cases}. \tag{6}$$

Hence, $\{Y(t)\}_{t\geq 0}$ is a discrete-time Markov chain such that each coordinate $\{Y_i(t)\}_{t\geq 0}$ evolves autonomously according to the transition probabilities given by the recurrence in (6). Each individual coordinate evolves according to a discrete-time Markov chain on a cycle of states $0, 1, \ldots, k-1$ where at each time instance the state of this Markov chain either increments modulo k with probability p_i or remains at the current state with probability $1 - p_i$. It is not difficult to see that the transition matrix of this Markov chain is doubly stochastic and irreducible (due to the requirement that $\min_i p_i > 0$), hence it has a unique stationary distribution which is uniform. The claim about the convergence follows from the fact that the aperiodicity of $X(t)$ extends to $Y(t)$, so that $Y(t)$ is an ergodic Markov chain.

□

Theorem 3.1. *The expected system latency satisfies* $\tau \geq \sqrt{k}/\|\mathbf{p}\|_2$*.*

Proof. Fix an arbitrary bin i. By the definition of the Markov chain $\{X(t)\}_{t\geq 0}$, we have

$$\mathbf{Pr}[X_i(t+1) = 0] = (1 - p_i)\mathbf{Pr}[X_i(t) = 0] + \sum_{j\neq i} p_i \mathbf{Pr}[X_j(t) = k, X_i(t) = k].$$

Hence, for the expected values with respect to the stationary distribution we have

$$p_i \mathbf{Pr}[X_i(t) = 0] = \sum_{j\neq i} p_j \mathbf{Pr}[X_j(t) = k, X_i(t) = k].$$

By Lemma 2.1, $\mathbf{Pr}[X_i(t) = 0] + \mathbf{Pr}[X_i(t) = k] = 1/k$. Therefore, we can rewrite the previous relation as

$$\sum_{j=1}^{n} p_j \mathbf{Pr}[X_i(t) = k, X_j(t) = k] = \frac{1}{k} p_i.$$

Using this identity, we obtain

$$\begin{aligned} \frac{1}{k}\sum_{i=1}^{n} p_i^2 &= \sum_{i=1}^{n} p_i \sum_{j=1}^{n} p_j \mathbf{Pr}[X_i(t) = k, X_j(t) = k] \\ &= \mathbf{E}\left[\left(\sum_{i=1}^{n} p_i \mathbf{1}_{X_i(t)=k}\right)^2\right] \\ &\geq \left(\sum_{i=1}^{n} p_i \mathbf{Pr}[X_i(t) = k]\right)^2 = \left(\sum_{i=1}^{n} \lambda_i\right)^2 = \lambda^2 \end{aligned}$$

where the last inequality is by Jensen's inequality and we use the fact that for every bin i, $\lambda_i = p_i \mathbf{Pr}[X_i(t) = k]$. We have thus showed that $\lambda \leq \sqrt{\sum_{i=1}^{n} p_i^2}/\sqrt{k} = \|\mathbf{p}\|_2/\sqrt{k}$. The proof is completed by recalling that $\tau = 1/\lambda$. □

Lemma 3.2. *For a random time t, the residual phase length $S(t)$ satisfies $\mathbf{E}[S(t)] = O(1/\|\mathbf{p}\|_2)$. Moreover, $\mathbf{Pr}[S(t) \leq C \cdot (\log n)^{1+\frac{\log k}{2}}/\|\mathbf{p}\|_2] \geq 1 - 1/n$, for a constant $C > 0$.*

Proof (continued). As in the first part of the proof, fix a level ℓ, whose occupancy at time t is given by the random variables $(Z_{\ell,i}(t))_{1 \leq i \leq n}$. By Theorem C.2, we have that the time until some bin at level ℓ acquires ℓ balls is $O(\log n / Z_\ell(t)^{1/\ell})$ with high probability, i.e., with probability at least $1 - n^{-1}$.

Recall from the expectation bound argument that, by Lemma 2.1 and Hoeffding's inequality, we obtain, for each level $2 \leq \ell \leq k$, a series of concentration bounds of the type:

$$Z_\ell^{1/\ell}(t) \leq \|\mathbf{p}\|_\ell/(2k)^{1/\ell} \text{ with probability at most } \exp\left(-\frac{1}{2k^2}\frac{\|\mathbf{p}\|_\ell^{2\ell}}{\|\mathbf{p}\|_{2\ell}^{2\ell}}\right).$$

Let j be the first level in $2 \leq \ell \leq k$, in ascending order, for which the inequality $\|\mathbf{p}\|_\ell^{2\ell}/\|\mathbf{p}\|_{2\ell}^{2\ell} \geq \log n$ holds, or $k+1$, if no such level exists. We have that the residual phase length is $O(\log n/\|\mathbf{p}\|_j)$, with high probability. To complete the proof, we need to relate $\|\mathbf{p}\|_j$ and $\|\mathbf{p}\|_2$.

For this, it is sufficient to notice that, by the definition of j, $\|\mathbf{p}\|_\ell/\|\mathbf{p}\|_{\ell+1} \leq (\log n)^{\frac{1}{2\ell}}$, for all $2 \leq \ell \leq j-1$. We can multiply out these inequalities to obtain that $\|\mathbf{p}\|_2 \leq \|\mathbf{p}\|_j (\log n)^{\frac{\log k}{2}}$. Putting together the last two steps, it follows that the residual phase length from t is $O((\log n)^{1+\frac{\log k}{2}}/\|\mathbf{p}\|_2)$, with high probability. Applying Lemma A.1 completes the proof of this claim. $\square$

Lemma A.1. *Let S_r be the length of a random phase r, and let $S(t)$ be the residual length of the phase at a random time t. Then the following relations hold: $\mathbf{E}[S_r] \leq 2\,\mathbf{E}[S(t)]$ and $\mathbf{Pr}[S_r > 2\theta] \leq 2\,\mathbf{Pr}[S(t) > \theta]$, for $\theta \geq 0$.*

Proof. The first claim is showed as follows. By the Palm inversion formula, we have

$$\begin{aligned}\mathbf{E}[S(t)] = \frac{\mathbf{E}[\sum_{s=0}^{S_r-1}(S_r - s)]}{\mathbf{E}[S_r]} &= \frac{\mathbf{E}[S_r(S_r+1)]}{2\,\mathbf{E}[S_r]} \\ &\geq \frac{\mathbf{E}[S_r^2]}{2\,\mathbf{E}[S_r]} \\ &\geq \frac{1}{2}\,\mathbf{E}[S_r]\end{aligned}$$

where the last inequality is by Jensen's inequality.

The second claim is showed as follows. Again, by the Palm inversion formula we have

$$\begin{aligned}\mathbf{Pr}[S(t) > \theta] &= \frac{\mathbf{E}[\sum_{s=0}^{S_r-1}\mathbf{1}_{S_r - s > \theta}]}{\mathbf{E}[S_r]} \\ &= \frac{\mathbf{E}[\max\{S_r - \theta, 0\}]}{\mathbf{E}[S_r]} \\ &\geq \frac{\mathbf{E}[\max\{S_r - \theta, 0\}\mathbf{1}_{S_r > 2\theta}]}{\mathbf{E}[S_r]} \\ &\geq \frac{\mathbf{E}[S_r \mathbf{1}_{S_r > 2\theta}]}{2\,\mathbf{E}[S_r]} \\ &\geq \frac{\mathbf{E}[S_r]\,\mathbf{E}[\mathbf{1}_{S_r > 2\theta}]}{2\,\mathbf{E}[S_r]} \\ &= \frac{1}{2}\mathbf{Pr}[S_r > 2\theta]\end{aligned}$$

where the last inequality follows by the Harris' inequality (Lemma D.5) becuase S_r is a real-valued random variable and $\mathbf{E}[S_r \mathbf{1}_{S_r > 2\theta}] = \mathbf{E}[f(S_r)g(S_r)]$ for two non-decreasing functions f and g defined by $f(x) = x$ and $g(x) = \mathbf{1}_{x > 2\theta}$ for $x \in \mathbf{R}$. $\square$

B Omitted Details from Section 4

Theorem 4.10. *For any bin i, the winning probability satisfies $w_i = \Omega(p_i^2/\|\mathbf{p}\|_2^2)$.*

Proof. By Lemma 3.2, there is a constant $c \geq 1$ so that the residual phase length satisfies $\mathbf{E}[S(t)] \leq c \cdot 1/\|\mathbf{p}\|_2$. We continue with a case distinction regarding p_i.

1. Case: $p_i \geq \|\mathbf{p}\|_2/(100ck^2)$. In this case, we conservatively assume that at the beginning of a phase, bin i has 0 balls and each other bin has $k-2$ balls. In addition, we will also assume that $p_i \leq 1/(100ck)$ because, otherwise, bin i has a constant winning probability and the statement holds trivially.

Consider now the balls that are allocated to the $n-1$ bins except bin i for the next $s := 1/\|\mathbf{p}\|_2$ times. For any bin $j \neq i$, let Z_j be a binary random variable which takes value 1 if bin j receives at least two balls in the given s time steps, and takes value 0, otherwise. Then, for $Z := \sum_{j\neq i} Z_j$ we have by linearity of expectation

$$\mathbf{E}[Z] \leq \sum_{j\neq i} \binom{s}{2} \cdot p_j^2 \leq \frac{1}{2},$$

and hence by Markov's inequality, $\mathbf{Pr}[Z \geq 1] \leq \mathbf{Pr}[Z \geq 2 \cdot \mathbf{E}[Z]] \leq 1/2$, which in turn implies $\mathbf{Pr}[Z = 0] \geq 1/2$.

On the other hand, the probability that bin i receives at least k balls during $1/(100cp_i) \leq s$ time-steps is at least some constant > 0, since $1/(100cp_i) \geq k$ and $1/p_i \leq (100ck^2)/\|\mathbf{p}\|_2$. Since these two events are negatively correlated, it follows that bin i wins with a constant probability > 0.

2. Case: $p_i \leq \|\mathbf{p}\|_2/(100ck^2)$. To lower bound the winning probability of bin i, we will first prove that in a random phase, bin i starts with $k-2$ balls with constant probability > 0. In order to establish this, for a random time t, we consider the following three events:

1. $\mathcal{E}_1$: bin i has $k-2$ balls at time t
2. $\mathcal{E}_2$: the residual phase length at time t is less than or equal to $(10ck)/\|\mathbf{p}\|_2$ time steps
3. $\mathcal{E}_3$: bin i does not get any ball during $(10ck)/\|\mathbf{p}\|_2$ time steps after time step t.

By Lemma 2.1, $\mathbf{Pr}[\mathcal{E}_1] = 1/k$, by Markov's inequality, $\mathbf{Pr}[\mathcal{E}_2] \geq 1 - 1/(10k)$ and by a simple union bound, $\mathbf{Pr}[\mathcal{E}_3] \geq 1 - (10ck/\|\mathbf{p}\|_2) \cdot p_i \geq 1 - 1/(10k)$. Another application of the union bound yields

$$\mathbf{Pr}[\mathcal{E}_1 \wedge \mathcal{E}_2 \wedge \mathcal{E}_3] \geq 1 - \frac{k-1}{k} - \frac{1}{10k} - \frac{1}{10k} = \frac{4}{5k}.$$

If all these three events occur, bin i has $k-2$ balls at the beginning of the next phase after time t.

The final step of this proof is somewhat similar to the proof of Lemma 4.5. In the following, we will call a time step t *good* if the event $\mathcal{E}_1 \wedge \mathcal{E}_2 \wedge \mathcal{E}_3$ occurs. Further, we will call a phase *good* if bin i has $k-2$ balls at the beginning of that phase. By the above considerations, it follows that for any time interval $[0, T]$, the number of good time-steps is $\Omega(T)$ with probability 1, as $T \to \infty$. Since every good phase corresponds to at most $10ck/\|\mathbf{p}\|_2$ good time steps, it follows that the number of good phases within $[0, T]$ is $\Omega(T/\|\mathbf{p}\|_2)$, w.p. 1 as $T \to \infty$. Finally, we know from Theorem 3.3 that the expected phase length is $\Theta(1/\|\mathbf{p}\|_2)$, hence the ratio of the number of good phases to the number of phases within a time interval of T time steps is $\Omega(1)$, w.p. 1 as $T \to \infty$. Combining these insights yields that a random phase is good with constant probability > 0. By Theorem C.3, bin i wins in every good phase with probability at least $\Omega(p_i^2/\|\mathbf{p}\|_2^2)$. Hence $w_i = \Omega(1) \cdot \Omega(p_i^2/\|\mathbf{p}\|_2^2) = \Omega(p_i^2/\|\mathbf{p}\|_2^2)$, as needed. □

Lemma 4.4. *For every level $2 \leq \ell \leq k$, a random time t is ℓ-perfect w. p. at least* $1 - \exp\Big(-\Omega\big((\frac{\|\mathbf{p}\|_\ell}{\|\mathbf{p}\|_{\ell+1}})^4\big)\Big)$.

Proof. Let us denote by $N_\ell(t)$ the set of bins which are at level ℓ, i.e., they need to acquire ℓ more balls to win the next phase. Further, let us define

$$\phi_\ell := \frac{\|\mathbf{p}\|_\ell}{\|\mathbf{p}\|_{\ell+1}} \geq 1.$$

By Lemma 2.1, we know that at a random time t the occupancies of bins k are independent and uniformly distributed. By applying Hoeffding's inequality (Lemma D.4),

$$\mathbf{Pr}\left[\sum_{i \in N_\ell(t)} p_i^\ell \geq \|\mathbf{p}\|_\ell^\ell/(2k)\right] \geq 1 - \exp\left(-\frac{2(\|\mathbf{p}\|_\ell^\ell/(2k))^2}{\|\mathbf{p}\|_{2\ell}^{2\ell}}\right) \geq 1 - \exp\left(-\frac{1}{4k^2}\phi_\ell^4\right). \tag{7}$$

We now consider the execution over the next $s := /(32k\|\mathbf{p}\|_2)$ time steps (independent of how many new phases start between times t and $t+s$). Let $N_\ell(t,s)$ be the set of bins in $N_\ell(t)$ which did not receive any ball in the time interval $[t, t+s-1]$; in other words, $N_\ell(t,s) = \cap_{i=0}^{s-1} N_\ell(t+i)$. Further, define

$$Z := \sum_{i \in N_\ell(t) \setminus N_\ell(t,s)} p_i^\ell$$

which is an upper bound on the ℓ-th norm of the probability mass which leaves the set $N_\ell(t)$ (at least once) until time $t+s$. Every bin $i \in N_\ell(t)$ is picked within the time-interval $[t, t+s-1]$ with probability at most $s \cdot p_i$, and hence by linearity of expectation,

$$\mathbf{E}[Z] \leq \sum_{i \in N_\ell(t)} p_i^\ell \cdot (s \cdot p_i) = \frac{\|\mathbf{p}\|_{\ell+1}^{\ell+1}}{32k\|\mathbf{p}\|_2} \leq \frac{\|\mathbf{p}\|_{\ell+1}^{\ell+1}}{32k\|\mathbf{p}\|_\ell} \leq \frac{1}{32k} \cdot \|\mathbf{p}\|_\ell^\ell.$$

In order to analyze Z, we consider a corresponding Poisson model which is run for a total time of s, and every bin i is associated with an independent Poisson process with rate p_i. Hence the number of times a bin i is picked until time s is a Poisson random variable with parameter $s \cdot p_i$, and the probability that bin i is picked at least once is at most $1 - e^{-s \cdot p_i} \leq s \cdot p_i$. Let

$$\widetilde{Z} := \sum_{i \in \widetilde{N}_\ell(t) \setminus \widetilde{N}_\ell(t,s)} p_i^\ell,$$

be the random variable corresponding to Z in the Poisson model. Then $\mathbf{E}[\widetilde{Z}] \leq \|\mathbf{p}\|_\ell^\ell/(32k)$, and by Hoeffding's inequality,

$$\mathbf{Pr}\left[\widetilde{Z} - \mathbf{E}[\widetilde{Z}] \geq \frac{1}{16k}\|\mathbf{p}\|_\ell^\ell\right] \leq \exp\left(-\frac{2(\frac{1}{16k}\|\mathbf{p}\|_\ell^\ell)^2}{\|\mathbf{p}\|_{2\ell}^{2\ell}}\right) \leq \exp\left(-\frac{1}{128k^2}\phi_\ell^4\right).$$

Furthermore, the random variable $\widetilde{Z}$ is non-increasing in s, which implies the following relation between the Poisson and the original process (cf. [19, Chapter 8]): For any $\theta \geq 0$,

$$\mathbf{Pr}[Z \geq \theta] \leq 4 \cdot \mathbf{Pr}[\widetilde{Z} \geq \theta].$$

Finally, this yields

$$\mathbf{Pr}\left[Z \geq \mathbf{E}[Z] + \|\mathbf{p}\|_\ell^\ell/(16k)\right] \leq 4 \cdot \exp\left(-\frac{1}{128k^2}\phi_\ell^4\right). \tag{8}$$

Since $\mathbf{E}[Z] + \|\mathbf{p}\|_\ell^\ell/(16k) \leq \|\mathbf{p}\|_\ell^\ell/(32k) + \|\mathbf{p}\|_\ell^\ell/(16k) \leq \|\mathbf{p}\|_\ell^\ell/(8k)$, taking the union bound over the events in equation (7) and equation (8) implies that with probability at least $1 - \exp\left(-\phi_\ell^4/(8k^2)\right) - 4 \cdot \exp\left(-\phi_\ell^4/(128k^2)\right) \geq 1 - 8 \cdot \exp\left(-\phi_\ell^4/(128k^2)\right)$, all time steps in the $[t, t+1/(\|32k\mathbf{p}\|_2)]$ are perfect. This completes the proof. □

Recall that in the following, the level $2 \leq \ell \leq k$ will be fixed so that we simply write perfect, good and bad instead of ℓ-perfect, ℓ-good and ℓ-bad.

Lemma 4.7. *Consider Process 2 running until the f-th epoch is completed, and let $B = B(f)$ and $L = L(f)$ be the values of the counters B and L at this time. Then the following two statements hold. First, $B(f)$ counts exactly the total number of bad phases that may occur until the last time step of the f-th epoch. Second, the expression $L(f) = \sum_{e=1}^{f} \big(\frac{1}{32k\|\mathbf{p}\|_2} + \sum_{j=1}^{|E_e|-1} |E_{e,j}| \big)$ lower bounds the number of non-perfect time steps which belong to a phase of the first f epochs.*

Proof. The first claim follows by the fact that every bad phase has to be part of an epoch.

For the second claim, consider the first phase $E_{e,1}$ of any epoch e and the most recent time-step t prior to that phase which is perfect. Then the time steps in the interval $I = [t+1, t+1/(32k\|\mathbf{p}\|_2)]$ are good, but not perfect. Furthermore, the time-step t cannot be part of the phase $E_{e,1}$ itself (as the first phase of an epoch is a bad phase), and moreover, if it is part of the phase of the previous epoch, then it has to be phase $E_{e-1,|E_{e-1}|}$. Furthermore, any other phase $E_{e,j}$ with $1 < j < |E_e|$ includes only non-perfect time-steps which can be also credited to epoch e. Adding up the contributions yields the second claim. □

Lemma 4.8. *Consider the original process and modified process both running until epoch f is completed, and let $B(f)$, $L(f)$, $\widetilde{B}(f)$, $\widetilde{L}(f)$ be the value of the four counters at this time. Then $L(f)/B(f) \succeq \widetilde{L}(f)/\widetilde{B}(f)$.*

Proof. We define our coupling in the natural way by assuming that both processes use identical sets of bin samples $M_{\ell,j} \in \{1,\ldots,n\}^{\mathbb{N}}$. Figure 2 summarizes the updates of the two counters in the modified process and the induced conditions on the updates of the two counters in the original process. These conditions follow from the definition of $\widetilde{S}_e$, since whenever epoch L_e contains more than $\widetilde{S}_e$ phases, then at least half of their phases take at least $1/(16k\|\mathbf{p}\|_2)$ steps. This implies that in at least $1/4$ of the phases, the counter L is incremented by $\frac{1}{32k\|\mathbf{p}\|_2}$ (recall that the last phase is not taken into account by L).

Case	Modified Process	Original Process	
$\widetilde{S}_e = 0$	$\widetilde{L} \leftarrow \widetilde{L} + \frac{1}{128k\|\mathbf{p}\|_2}$ $\widetilde{B} \leftarrow \widetilde{B} + 1$	$L \leftarrow L + \Delta L$ $B \leftarrow B + \Delta B$	s.t. $\frac{\Delta L}{\Delta B} \geq \frac{1}{128k\|\mathbf{p}\|_2}$ and $\Delta B \geq 1$
$\widetilde{S}_e \neq 0$	$\widetilde{L} \leftarrow \widetilde{L} + 0$ $\widetilde{B} \leftarrow \widetilde{B} + \widetilde{S}_e$	$L \leftarrow L + \Delta L$ $B \leftarrow B + \Delta B$	s.t. $\begin{cases} 1 \leq \Delta B < \widetilde{S}_e, \Delta L = 0, \text{ or} \\ \frac{\Delta L}{\Delta B} \geq \frac{1}{128k\|\mathbf{p}\|_2}. \end{cases}$

Figure 2: Update rules of the original and the modified process for epoch E_e.

Next we observe that we may replace the condition $\Delta L/\Delta B \geq 1/(128k\|\mathbf{p}\|_2)$ by $\Delta L/\Delta B = 1/(128k\|\mathbf{p}\|_2)$, since this can only decrease $L(f)/B(f)$. For the same reason, we may also replace the condition $1 \leq \Delta B < \widetilde{S}_e$ by $\Delta B = \widetilde{S}_e$.

Then, let $A(f)$ be the set of indices $1 \leq e \leq f$ with $\widetilde{S}_e = 0$. With this notation, the modified process produces a ratio of

$$\frac{\widetilde{L}(f)}{\widetilde{B}(f)} = \frac{\sum_{i \in A(f)} \frac{1}{128k\|\mathbf{p}\|_2}}{\sum_{i \in A(f)} 1 + \sum_{e \notin A(f)} \widetilde{S}_e}.$$

For the original process, we can find a scaling parameter $\beta = \beta(f) > 0$ so that

$$\frac{L(f)}{B(f)} \geq \frac{\beta \cdot \frac{1}{128k\|\mathbf{p}\|_2}}{\beta + \sum_{e \notin A(f) \colon \Delta B \leq \widetilde{S}_e} \widetilde{S}_e}.$$

Now using the fact that the last expression is non-decreasing in β,

$$\frac{L(f)}{B(f)} \geq \frac{|A(f)| \cdot \frac{1}{128k\|\mathbf{p}\|_2}}{|A(f)| + \sum_{e \notin A(f) \colon \Delta B \leq \widetilde{S}_e} \widetilde{S}_e} \geq \frac{|A(f)| \cdot \frac{1}{128k\|\mathbf{p}\|_2}}{|A(f)| + \sum_{e \notin A(f)} \widetilde{S}_e} = \frac{\widetilde{L}(f)}{\widetilde{B}(f)}.$$

Hence there is a coupling so that

$$\mathbf{Pr}\left[\frac{L(f)}{B(f)} \geq \frac{\widetilde{L}(f)}{\widetilde{B}(f)}\right] = 1,$$

which implies the desired majorization. $\square$

Lemma 4.9. *For the modified process (Process 3), the following statement holds.*

- *For any $\theta \geq 1$ and epoch e, $\mathbf{Pr}[\widetilde{S}_e \geq \theta] \leq (1/4)2^{-\theta}$, thus, in particular, $\mathbf{Pr}[\widetilde{S}_e < \infty] = 1$.*
- *Let $\widetilde{B} = \widetilde{B}(f)$ and $\widetilde{L} = \widetilde{L}(f)$ be the values of the counters $\widetilde{B}$ and $\widetilde{L}$ until epoch f is completed. Then, $\widetilde{L}(f) \succeq \mathsf{Bin}(f, 1/2)/(128k\|\mathbf{p}\|_2)$, and $\widetilde{B}(f) \preceq \mathsf{Geo}(1/2) \cdot f$.*

Proof. First, consider an arbitrary epoch e. Recall that $Z_{e,j} = 1$ if and only if there is no bin which gets two balls in the j-th phase of the e-th epoch during $1/(16k\|\mathbf{p}\|_2)$ steps. Hence if $Z_{e,j} = 1$, then this phase will take $1/(16k\|\mathbf{p}\|_2)$ steps, regardless of the initial occupancy configuration at the beginning of that phase. Then for any $j \geq 1$, it follows by Theorem C.2 with $\delta = (1/16)^2$ and $k = 2$, $\mathbf{Pr}[Z_{e,j} = 0] \leq \frac{1}{256}$. Therefore, for any $\theta \geq 1$,

$$\mathbf{Pr}[\widetilde{S}_e \geq \theta] \leq \sum_{k=\theta}^{\infty} \binom{k}{k/2} 256^{-k/2} \leq \sum_{k=\theta}^{\infty} 2^k \cdot 2^{-4k} = \sum_{k=\theta}^{\infty} 2^{-3k} \leq \frac{1}{4} \cdot 2^{-\theta}, \tag{9}$$

which establishes the first claim. The second claim about $\widetilde{L}(f)$ follows directly, as the first claim implies $\mathbf{Pr}[\widetilde{S}_e = 0] \geq 1/2$ for any epoch e, and these events are independent across epochs $e = 1, 2, \ldots, f$. The claim about $\widetilde{B}(f)$ follows since the tails in equation (9) are smaller than the corresponding ones for $\mathsf{Geo}(1/2)$. $\square$

Completing the Proof of Theorem 4.1

We will now complete the proof of Theorem 4.1 that the winning probability of bin i is at most $O(p_i^2/\|\mathbf{p}\|_2^2)$. The final step of this proof is somewhat to similar to the one of Lemma 3.2.

Recall that N_j^t denotes the set of bins which have load $k - j$ at time-step t (i.e., they need to aquire j more balls to win the next phase).

Proof of Theorem 4.1. For $\ell \in \{2, 3, \ldots, k\}$, let $\mathcal{E}_\ell$ be the event that a random phase is ℓ-good. Given an arbitrary occupancy configuration, let $\mathcal{E}_j$ be the first event of this type that holds, where we consider events in increasing order of ℓ. Therefore, the norm bounds for all previous levels have failed, but the current bound is correct. Ignoring all balls which are allocated to bins outside N_j^t and bin i, we only increase the winning probability of bin i. Formally, we consider a different probability vector $\widetilde{p}$ defined by $\widetilde{p}_j := p_j \cdot \beta$, where $\beta := 1/(p_i + \sum_{k \in N_j(t)} p_k)$ is a scaling factor to ensure that $\|\widetilde{p}_j\|_1 = 1$. Applying Corollary C.4 to $\widetilde{p}$ implies that the probability for bin i to win is still at most

$$O\left(\frac{\widetilde{p}_i^2}{\left(\sum_{i \in N_j(t)} \widetilde{p}_i^j\right)^{2/j}}\right) = O\left(\frac{p_i^2}{\left(\sum_{i \in N_j(t)} p_i^j\right)^{2/j}}\right) = O\left(\frac{p_i^2}{\|\mathbf{p}\|_j^2}\right),$$

where the first equality holds since $\widetilde{p}$ is obtained from p by scaling, and the second inequality holds since the event $\mathcal{E}_j$ holds.

We then obtain that the following upper bound on the winning probability holds, up to a constant.

$$\frac{p_i^2}{\|\mathbf{p}\|_2^2} + \sum_{\ell=2}^{k} \mathbf{Pr}\left[\overline{\mathcal{E}_2} \wedge \overline{\mathcal{E}_3} \wedge \ldots \wedge \overline{\mathcal{E}_\ell}\right] \frac{p_i^2}{\|\mathbf{p}\|_\ell^2}. \tag{10}$$

We now focus on bounding the probability term for fixed ℓ. By Lemma 4.3,

$$\mathbf{Pr}\left[\overline{\mathcal{E}_j}\right] \leq \exp\left(-c \cdot \frac{\|\mathbf{p}\|_j^4}{\|\mathbf{p}\|_{j+1}^4}\right),$$

where $c > 0$ is some constant, which implies the following:

$$\mathbf{Pr}\left[\overline{\mathcal{E}_2} \wedge \overline{\mathcal{E}_3} \wedge \ldots \wedge \overline{\mathcal{E}_\ell}\right] \leq \prod_{j=2}^{\ell} \exp\left(-\frac{c}{2j} \frac{\|\mathbf{p}\|_j^4}{\|\mathbf{p}\|_{j+1}^4}\right).$$

This holds since we can bound the conjunction by the least probable event, and, in turn, this event is less likely than the ℓth root of the product of probabilities for all events. In turn, we have

$$\prod_{j=2}^{\ell} \exp\left(-\frac{c}{2j} \frac{\|\mathbf{p}\|_j^2}{\|\mathbf{p}\|_{j+1}^2}\right) \leq \prod_{j=2}^{\ell} \frac{2j}{c} \frac{\|\mathbf{p}\|_{j+1}^2}{\|\mathbf{p}\|_j^2} \leq \frac{2\ell!}{c} \frac{\|\mathbf{p}\|_{\ell+1}^2}{\|\mathbf{p}\|_2^2},$$

where in the last step we have used the fact that $\exp(-x) \leq 1/x$ for $x > 0$. Returning to Equation 10, we obtain that there exists a constant $C > 0$ such that the phase length is upper bounded by

$$C\left(\frac{p_i^2}{\|\mathbf{p}\|_2^2} + \sum_{\ell=2}^{k} \frac{\|\mathbf{p}\|_{\ell+1}^2}{\|\mathbf{p}\|_2^2} \cdot \frac{p_i^2}{\|\mathbf{p}\|_\ell^2}\right) = O\left(\frac{p_i^2}{\|\mathbf{p}\|_2^2}\right).$$

□

C Analysis of a Simpler Balls-into-Bins Process

For the analysis of our original balls-into-bins process $\{X(t)\}_{t\geq 0}$, we will require latency bounds for a simpler balls-into-bins process in which *all* bins restart to one at an attempt to add a ball into a bin with k balls. Hence unlike the original process $\{X(t)\}_{t\geq 0}$, the configuration at the beginning of every phase is identical, which implies the following simple but useful lemma.

Lemma C.1 (Sub-Multiplicativity). *For any two integers $\theta \geq 1, y \geq 1$, the length of an arbitrary phase r, S_r, satisfies*

$$\mathbf{Pr}[S_r \geq y \cdot \theta] \leq \mathbf{Pr}[S_r \geq \theta]^y.$$

Proof. This lemma follows by simply dividing the time-interval of length $y \cdot \theta$ into y consecutive time-intervals of length θ. Then, for any of these fixed intervals of length θ, the probability that there is no bin which gets k balls within this interval equals $\mathbf{Pr}[S_r \geq \theta]$. □

Theorem 2.2. *The length S_r of a random phase r and the probabilities w_i that bin i wins an arbitrary phase, in the process where all bins reset to one ball on a win, satisfy:*

$$\mathbf{E}[S_r] \in \left[\frac{(k!/2)^{1/k}}{2\|\mathbf{p}\|_k}, \frac{8ek^k}{\|\mathbf{p}\|_k}\right] \text{ and } w_i = \Theta\left(\frac{p_i^k}{\|\mathbf{p}\|_k^k}\right) \text{ for every bin } 1 \leq i \leq n.$$

We split the proof of this theorem into two parts, the analysis of phase length (system latency) and the analysis of winning probabilities.

C.1 Analysis of Phase Length

In the next theorem we provide bounds for the probability of deviation and the expected value of the length of a random phase r. In particular, this theorem implies that for the simple balls-into-bins game the expected phase length satisfies $\mathbf{E}[S_r] = \Theta(1/\|\mathbf{p}\|_k)$.

Theorem C.2. *The length of an arbitrary phase r, S_r, satisfies the following properties:*

- *Lower bound:*

$$\mathbf{Pr}\left[S_r > \frac{(\delta k!)^{1/k}}{\|\mathbf{p}\|_k}\right] \geq 1-\delta, \textit{ for } 0<\delta<1$$

and

$$\mathbf{E}[S_r] \geq \frac{1}{2}\frac{(k!/2)^{1/k}}{\|\mathbf{p}\|_k}.$$

- *Upper bound:*

$$\mathbf{Pr}\left[S_r \leq \frac{1}{\|\mathbf{p}\|_k}\right] \geq \frac{1}{4ek^k}$$

and

$$\mathbf{E}[S_r] \leq \frac{4ek^k}{\|\mathbf{p}\|_k}.$$

The bounds in Theorem C.2 even apply to vectors $\mathbf{p} = (p_1, p_2, \ldots, p_n) \in \mathbf{R}_+^n$ such that $0 < \sum_{i=1}^n p_i < 1$, which as outlined Section 2, corresponds to a lazy version where no ball is placed with probability $1 - \sum_{i=1}^n p_i$.

Proof. We first prove the two lower bounds. We shall assume for simplicity that the phase begins at time-step 1. Let Y be the number of bins which receive at least k balls over t time steps from the beginning of an arbitrary phase r. By linearity of expectations and the fact that for any $Z \sim \mathsf{Bin}(m,p)$, $\mathbf{Pr}[Z \geq s] \leq \binom{m}{s} \cdot p^s$, we have

$$\mathbf{E}[Y] \leq \sum_{i=1}^n \binom{t}{k} \cdot p_i^k \leq \frac{t^k}{k!} \cdot \|\mathbf{p}\|_k^k.$$

Choosing $t = (\delta k!)^{1/k}/\|\mathbf{p}\|_k$ yields $\mathbf{E}[Y] \leq \delta$, and by Markov's inequality,

$$\mathbf{Pr}[S_r \leq t] = \mathbf{Pr}[Y \geq 1] \leq \mathbf{Pr}\left[Y \geq \frac{1}{\delta} \cdot \mathbf{E}[Y]\right] \leq \delta.$$

Consequently, for $\delta = 1/2$ and $t = (k!/2)^{1/k}/\|\mathbf{p}\|_k$,

$$\mathbf{E}[Y] \geq \mathbf{Pr}[S_r \geq t] \cdot t \geq \frac{1}{2} \cdot t.$$

In order to prove the two upper bounds, we first lower bound $\mathbf{E}[Y]$ and then do a second moment analysis. First,

$$\mathbf{E}[Y] \geq \sum_{i=1}^n \binom{t}{k} \cdot p_i^k \cdot (1-p_i)^{t-k} \geq \frac{t^k}{k^k} \cdot \sum_{i=1}^n p_i^k \cdot (1-p_i)^{t-k}.$$

We now choose $t := 1/\|\mathbf{p}\|_k$. In particular, this implies $t \leq 1/p_i$ and using the fact (cf. Lemma D.2) that $(1-p_i)^{t-k} \geq (1-p_i)^{1/p_i - 1} \geq 1/e$, we conclude

$$\mathbf{E}[Y] \geq \frac{t^k}{e \cdot k^k} \cdot \|\mathbf{p}\|_k^k = \frac{1}{e \cdot k^k}.$$

We now turn to $\mathbf{E}[Y^2]$. Notice that $Y = \sum_{i=1}^{n} Z_i$, where Z_i is a binary random variable taking value 1 if bin i has received at least k balls in the given interval of t time steps, and taking value 0 otherwise. Hence,

$$\begin{aligned}
\mathbf{E}[Y^2] &= \mathbf{E}\left[\Big(\sum_{i=1}^{n} Z_i\Big)^2\right] \\
&= \sum_{i=1}^{n} \mathbf{E}[Z_i^2] + \sum_{1 \le i \ne j \le n} \mathbf{E}[Z_i \cdot Z_j] \\
&= \mathbf{E}[Y] + \sum_{1 \le i \ne j \le n} \mathbf{Pr}[Z_i = 1] \cdot \mathbf{Pr}[Z_j = 1 \mid Z_i = 1]
\end{aligned}$$

where the last equality holds because $Z_i^2 = Z_i$. Next observe that $\mathbf{Pr}[Z_j = 1 \mid Z_i = 1] \le \mathbf{Pr}[Z_j = 1]$ which yields

$$\begin{aligned}
\mathbf{E}[Y^2] &\le \mathbf{E}[Y] + \sum_{1 \le i,j \le n} \mathbf{Pr}[Z_i = 1] \cdot \mathbf{Pr}[Z_j = 1] \\
&\le \mathbf{E}[Y] + \sum_{i=1}^{n} \mathbf{Pr}[Z_i = 1] \sum_{j=1}^{n} \mathbf{Pr}[Z_j = 1] \\
&\le \mathbf{E}[Y] + \mathbf{E}[Y]^2.
\end{aligned}$$

By Lemma D.3,

$$\mathbf{Pr}\left[Y \ge \frac{1}{2} \cdot \mathbf{E}[Y]\right] \ge \frac{1}{4} \cdot \frac{\mathbf{E}[Y]^2}{\mathbf{E}[Y^2]} \ge \frac{1}{4} \cdot \frac{\mathbf{E}[Y]^2}{\mathbf{E}[Y] + \mathbf{E}[Y]^2} = \frac{1}{4} \cdot \frac{\mathbf{E}[Y]}{1 + \mathbf{E}[Y]} \ge \frac{1}{8e \cdot k^k} := c_k,$$

where c_k is a strictly positive constant. This implies that phase r ends after $t = 1/\|\mathbf{p}\|_k$ time steps with probability at least $c_k > 0$.

Next recall that S_r is submultiplicative (cf. Lemma C.1), which means

$$\mathbf{Pr}[S_r \ge \theta \cdot t] \le \left(\mathbf{Pr}[S_r \ge t]\right)^{\theta}, \text{ for any } \theta \ge 1.$$

This in turn implies that S_r is stochastically smaller than a geometric random variable with parameter c_k/t, hence

$$\mathbf{E}[S_r] \le \frac{1}{c_k} \cdot t = \frac{8e \cdot k^k}{\|\mathbf{p}\|_k}.$$

□

C.2 Analysis of Winning Probabilities

Recall that w_i denotes the probability that bin i wins in an arbitrary phase.

Theorem C.3. *For every bin i, $w_i = \Theta\left(p_i^k / \|\mathbf{p}\|_k^k\right)$.*

Proof. **(Lower Bound).** Consider a random phase r that start at a time-step T_r, when all bins have zero balls. Let us define S_r^{-i} as the elapsed time until the first bin $\ne i$ reaches k balls, and let $t := 1/\|\mathbf{p}\|_k$. Applying Theorem C.2 with $\delta = (1/k!)$, $\mathbf{Pr}[S_r > t] \ge 1 - 1/k! \ge 1/2$. Since S_r^{-i} is stochastically larger than S_r, we have

$$\mathbf{Pr}\left[S_r^{-i} > t\right] \ge \mathbf{Pr}\left[S_r > t\right] \ge \frac{1}{2}.$$

Now define two events:

- $\mathcal{E}_1 := \{\text{bin } i \text{ has at least } k \text{ balls within } t \text{ time steps}\}$

- $\mathcal{E}_2 := \left\{ S_r^{-i} > t \right\}$

Observe that $\mathbf{Pr}[\mathcal{E}_2 \mid \mathcal{E}_1] \geq \mathbf{Pr}[\mathcal{E}_2]$, since conditioning on bin i having at least k balls at time t makes it more likely that none of the other $n-1$ bins has at least k balls at the same time. Therefore,

$$w_i \geq \mathbf{Pr}[\mathcal{E}_1 \wedge \mathcal{E}_2] \geq \mathbf{Pr}[\mathcal{E}_1] \cdot \mathbf{Pr}[\mathcal{E}_2] \geq \binom{t}{k} \cdot p_i^k \cdot (1-p_i)^{t-k} \cdot \frac{1}{2},$$

and since we have already shown in the proof of Theorem C.2 that $(1-p_i)^{1/\|\mathbf{p}\|_k - k} \geq 1/e$, it follows that

$$w_i \geq \frac{1}{k^k \cdot e \cdot 2} \cdot \frac{p_i^k}{\|\mathbf{p}\|_k^k}.$$

(Upper Bound). We first focus on the case $w_i \leq 1/4$ and defer the case $1/4 < w_i \leq 1$, which turns out to be easier, to the end of the proof. Let $\mathcal{W}_i$ denote the event that bin i wins, so $w_i = \mathbf{Pr}[\mathcal{W}_i]$. Let $\widetilde{S}_r^{-i}$ be the waiting time for a process where bin i has probability 0 and all other probabilities are increased by a factor of $1/(1-p_i)$. We can couple two executions of $\widetilde{S}_r^{-i}$ and S_r so that the subsequence of chosen bins $\neq i$ in S_r is equal to the sequence of chosen bins in $\widetilde{S}_r^{-i}$. Hence the random variable $\widetilde{S}_r^{-i}$ will be smaller than S_r, conditional on the event that bin i does not win. Hence,

$$\mathbf{Pr}[\widetilde{S}_r^{-i} \geq 4 \cdot \mathbf{E}[S_r] \mid \overline{\mathcal{W}_i}] \leq \mathbf{Pr}[S_r^{-i} \geq 4 \cdot \mathbf{E}[S_r] \mid \overline{\mathcal{W}_i}] \leq \frac{1}{4},$$

where the second inequality follows by Markov's inequality. This implies

$$\begin{aligned}\mathbf{Pr}[\widetilde{S}_r^{-i} \geq 4 \cdot \mathbf{E}[S_r]] &\leq (1-w_i) \cdot \mathbf{Pr}[\widetilde{S}_r^{-i} \geq 4 \cdot \mathbf{E}[S_r] \mid \overline{\mathcal{W}_i}] + w_i \cdot \mathbf{Pr}[\widetilde{S}_r^{-i} \geq 4 \cdot \mathbf{E}[S_r] \mid \mathcal{W}_i] \\ &\leq 1 \cdot \frac{1}{4} + \frac{1}{4} \cdot 1 = \frac{1}{2},\end{aligned}$$

where the second inequality follows by Markov's inequality and the assumption $w_i \leq 1/4$. Since the random variable $\widetilde{S}_r^{-i}$ is submultiplicative (cf. Lemma C.1), the above inequality implies that $\widetilde{S}_r^{-i}$ is stochastically smaller than a geometric random variable with parameter $1/2$ times $4\,\mathbf{E}[S_r]$ which yields

$$\mathbf{E}[\widetilde{S}_r^{-i}] \leq \frac{4\,\mathbf{E}[S_r]}{1-1/2} = 8 \cdot \mathbf{E}[S_r].$$

By Markov's inequality,

$$\mathbf{Pr}[\widetilde{S}_r^{-i} \geq 16\,\mathbf{E}[S_r]] \leq 1/2,$$

and by submultiplicativity,

$$\mathbf{Pr}[\widetilde{S}_r^{-i} \geq 16\theta \cdot \mathbf{E}[S_r]] \leq 1/2^\theta, \text{ for every integer } \theta. \tag{11}$$

It follows that

$$\begin{aligned}w_i &= \mathbf{Pr}[\text{bin } i \text{ wins before time } 32k \cdot \mathbf{E}[S_r]] + \sum_{s=32k\cdot\mathbf{E}[S_r]}^{\infty} \mathbf{Pr}[\text{bin } i \text{ wins within } s \text{ time steps}] \\ &\leq \mathbf{Pr}[\text{bin } i \text{ has } k \text{ balls before } 32k \cdot \mathbf{E}[S_r] \text{ time steps elapsed}] \\ &\quad + \sum_{s=32k\cdot\mathbf{E}[S_r]}^{\infty} \mathbf{Pr}[\text{bin } i \text{ has } k-1 \text{ balls before time } s] \cdot \mathbf{Pr}[\widetilde{S}_r^{-i} > s-k] \cdot p_i \\ &\leq \binom{32k \cdot \mathbf{E}[S_r]}{k} \cdot p_i^k + \sum_{s=32k\cdot\mathbf{E}[S_r]}^{\infty} \binom{s-1}{k-1} p_i^{k-1} \cdot \mathbf{Pr}[\widetilde{S}_r^{-i} > s/2] \cdot p_i,\end{aligned}$$

where in the first inequality we used our coupling above between S_r and S_r^{-i} to conclude that

$$\mathbf{Pr}[S_r > s-1 \mid \text{bin } i \text{ has } k-1 \text{ balls before } s \text{ time steps elapsed}] = \mathbf{Pr}[\widetilde{S}_r^{-i} > s-k].$$

Now we apply inequality (11) to obtain

$$\begin{aligned} w_i &= O\left(p_i^k \cdot \mathbf{E}[S_r]^k + p_i^k \cdot \sum_{s=32k\cdot \mathbf{E}[S_r]}^{\infty} s^{k-1} \cdot 2^{-\lfloor \ell/(32\,\mathbf{E}[S_r]) \rfloor}\right) \\ &= O\left(p_i^k \cdot \mathbf{E}[S_r]^k + p_i^k \cdot 32\,\mathbf{E}[S_r] \sum_{s=0}^{\infty} (32(s+1)k\,\mathbf{E}[S_r])^{k-1} \cdot 2^{-s}\right) \\ &= O\left(p_i^k \cdot \mathbf{E}[S_r]^k + p_i^k \cdot \mathbf{E}[S_r]^k\right) = O\left(\frac{p_i^k}{\|\mathbf{p}\|_k^k}\right), \end{aligned}$$

where the last line we used the bound $\mathbf{E}[S_r] = O(1/\|\mathbf{p}\|_k)$ from Theorem C.2.

Finally, let us consider the case $w_i \geq 1/4$. In this case,

$$w_i \leq \mathbf{Pr}[\text{bin } i \text{ gets } k \text{ balls before time } 8 \cdot \mathbf{E}[S_r]\,] + \mathbf{Pr}[S_r \geq 8 \cdot \mathbf{E}[S_r]] \leq O\left(\frac{p_i^k}{\|\mathbf{p}\|_k^k}\right) + \frac{1}{8},$$

and rearranging implies $w_i = O\left(p_i^k/\|\mathbf{p}\|_k^k\right)$, as needed. □

Corollary C.4. *Fix any bin $1 \leq i \leq n$ and consider the variant of the iterative balls-into-bins process where after every reset, bin i starts at level $k-2$ and all other bins start at level 1. Then bin i wins a random phase with probability at most $O(p_i^2/\|\mathbf{p}\|_k^2)$.*

Proof. The proof is almost identical to the proof of Theorem C.3, the only difference is that every power of k has to be replaced with a power of 2. □

D Useful Relations

Lemma D.1 (Palm inversion formula)**.** *Let $X(t)$ be a regenerative process (e.g. see [5]) on the state space E with regeneration points $\cdots < T_{-1} < T_0 = 0 < T_1 < \cdots$ and the let $S_r = T_{r+1} - T_r$ for $r \in \mathbf{Z}$. Suppose that $f : E \to \mathbf{R}$ is such that*

$$\mathbf{E}\left[\sum_{t=0}^{S_0-1} |f(X(t))|\right] < \infty.$$

Suppose π is the stationary distribution of the process $X(t)$. Then, we have

$$\sum_{\mathbf{x}\in E} f(\mathbf{x})\pi(\mathbf{x}) = \frac{\mathbf{E}\left[\sum_{t=0}^{S_0-1} f(X(t))\right]}{\mathbf{E}[S_0]}.$$

If the distribution of S_0 is proper and non-lattice, then for any initial state $\mathbf{x} \in E$,

$$\lim_{t\to\infty} \mathbf{E}[f(X(t)) \mid X(0) = \mathbf{x}] = \sum_{\mathbf{x}\in E} f(\mathbf{x})\pi(\mathbf{x}).$$

Lemma D.2. *For all $t, n \in \mathbb{R}$, such that $n \geq 1$ and $|t| \leq n$,*

$$e^t\left(1 - \frac{t^2}{n}\right) \leq \left(1 + \frac{t}{n}\right)^n \leq e^t.$$

Lemma D.3 (Paley-Zigmund's inequality)**.** *Let $Z \geq 0$ be any random variable. Then for any $0 < \theta < 1$,*

$$\mathbf{Pr}[Z \geq \theta \cdot \mathbf{E}[Z]] \geq (1-\theta)^2 \cdot \frac{\mathbf{E}[Z]^2}{\mathbf{E}[Z^2]}.$$

Lemma D.4 (Hoeffding's Inequality: one-sided versions))**.** *Let $X_1, X_2, \ldots, X_n$ be a sequence of random variables such that $X_i \in [a_i, b_i]$ for $1 \leq i \leq n$ with probability 1 and let $X := \sum_{i=1}^n X_i$. Then, it holds for any $\theta > 0$,*

$$\mathbf{Pr}[X - \mathbf{E}[X] \geq \theta] \leq \exp\left(-\frac{2\theta^2}{\sum_{i=1}^n (b_i - a_i)^2}\right),$$

and

$$\mathbf{Pr}[X - \mathbf{E}[X] \leq -\theta] \leq \exp\left(-\frac{2\theta^2}{\sum_{i=1}^n (b_i - a_i)^2}\right).$$

Lemma D.5 (Harris' inequality)**.** *Let X be a real-valued random variable. Then, for every pair of non-decreasing functions $f : \mathbf{R} \to \mathbf{R}$ and $g : \mathbf{R} \to \mathbf{R}$,*

$$\mathbf{E}[f(X)g(X)] \geq \mathbf{E}[f(X)]\,\mathbf{E}[g(X)].$$

E Notation Glossary

- n number of bins
- i denotes a bin
- t random time
- r random phase
- k the maximum occupancy of a bin
- $X_i(t) \in \{0, 1, \ldots, k\}$ number of balls in bin i at time t
- $\ell \in \{1, \ldots, k+1\}$ level of a bin (i.e., ℓ more balls are needed to win)
- $N_\ell(t)$ set of bins with level ℓ at time t
- S_r length of phase r
- $S(t)$ residual length of the phase at time t
- τ system latency
- τ_i latency of process i
- λ rate
- λ_i rate for bin i
- w_i the winning probability for bin i in a random phase
- E_e the e-th epoch (epochs are indexed $1, 2, \ldots$)
- $|E_e|$ size of epoch e, i.e., number of phases it contains
- $E_{e,j}$ the j-th phase of the e-th epoch
- $|E_{e,j}|$ length of the j-th phase of the e-th epoch
- B counter for bad phases
- L counter for non-perfect time steps

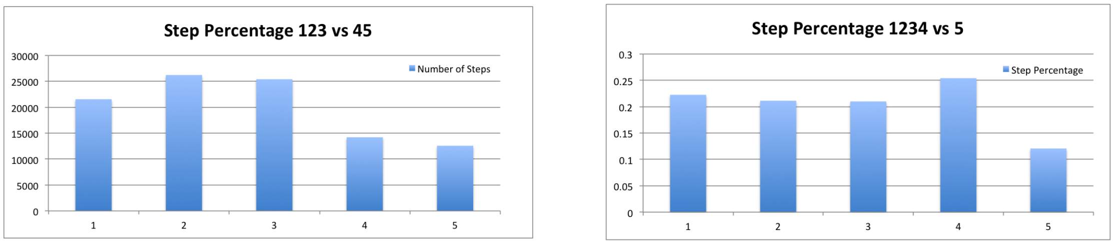

Figure 3: Example of how non-uniformity in the memory access times leads to non-uniform scheduling distribution.

F Empirical Evidence for Non-Uniformity

In this section, we provide some basic empirical evidence for the non-uniformity of stochastic schedulers on modern concurrent architectures. Such evidence was already provided by Dice et al. [7] for the case of single-socket machines, where all the cores are located on the same silicon die, as a consequence of cache effects, and is folklore for the case of multi-socket non-uniform memory access (NUMA) machines.

We provide a simple experiment exhibiting scheduling non-uniformity for a NUMA machine. More precisely, we consider a Fujitsu PRIMERGY RX600 S6 server with four Intel Xeon E7-4870 (Westmere EX) processors. Each processor has 10 2.40 GHz cores. Each core has private write-back L1 and L2 caches; an inclusive L3 cache is shared by all cores.

In our experiment, we run a simple application where five threads implement a shared counter using a lock-free pattern. In particular, each thread reads a shared variable, and then attempts to increment it via a compare-and-swap operation. We pin the five threads in different configurations across processors, so that not all threads are instantiated on the same processor, and observe the percentage of steps that each thread gets to take on the variable in relation to other threads over a one second period.

Figure 3 provides the results for two different configurations. In the first, we place threads 1, 2, 3, 4 on the same processor, which is also the same processor to which the memory location belongs, and thread 5 on an outside processor. As a consequence, threads 1 to 4 will have lower memory access times than 5, and in particular are scheduled to access memory about twice more often than 5. The second graph is the percentage of steps taken in the configuration where threads 1, 2, and 3 are on the first processor, and 4 and 5 are on another processor. The scheduling ratios of 4 and 5 are decreased as a consequence.