Self-Stabilizing Repeated Balls-into-Bins

L. Becchetti[1], A. Clementi[2], E. Natale[1], F. Pasquale[2], and G. Posta[1]

[1]*Sapienza* Università di Roma, becchett@dis.uniroma1.it, natale@di.uniroma1.it, gustavo.posta@mat.uniroma1.it
[2]Università *Tor Vergata* di Roma, clementi@mat.uniroma2.it, pasquale@mat.uniroma2.it

January 20, 2015

Abstract

We study the following synchronous process that we call *repeated balls-into-bins*. The process is started by assigning n balls to n bins in an arbitrary way. Then, in every subsequent round, one ball is chosen according to some fixed strategy (random, FIFO, etc) from each non-empty bin, and re-assigned to one of the n bins uniformly at random. This process corresponds to a non-reversible Markov chain and our aim is to study its *self-stabilization* properties with respect to the *maximum (bin) load* and some related performance measures.

We define a configuration (i.e., a state) *legitimate* if its maximum load is $O(\log n)$. We first prove that, starting from any legitimate configuration, the process will only take on legitimate configurations over a period of length bounded by *any* polynomial in n, *with high probability* (w.h.p.). Further we prove that, starting from *any* configuration, the process converges to a legitimate configuration in linear time, w.h.p. This implies that the process is self-stabilizing w.h.p. and, moreover, that every ball traverses all bins in $O(n \log^2 n)$ rounds, w.h.p.

The latter result can also be interpreted as an almost tight bound on the *cover time* for the problem of *parallel resource assignment* in the complete graph.

1 Introduction

We study the following *repeated balls-into-bins* process. Given any $n \geqslant 2$, we initially assign n balls to n bins in an arbitrary way. Then, at every round, from each non-empty bin one ball is chosen according to some strategy (random, FIFO, etc) and re-assigned to one of the n bins uniformly at random. Every ball thus performs a sort of *delayed* random walk over the bins and the delays of such random walks depend on the size of the bin queues encountered during their paths. It thus follows that these random walks are correlated. We study the impact of such correlation on the maximum load.

Inspired by previous concepts of (load) stability [1,7], we study the *maximum load* $M^{(t)}$, i.e., the maximum number of balls inside one bin at round t and we are interested in the largest $M^{(t)}$ achieved by the process over a period of *any polynomial* length. We say that a configuration is *legitimate* if its maximum load is $O(\log n)$ and a process is *stable* if, starting from any legitimate configuration, it only takes on legitimate configurations over a period of poly(n) length, w.h.p.

We also investigate a probabilistic version of self-stabilization [13]: we say that a process is *self-stabilizing*[1] if it is stable and if, moreover, starting from *any* configuration, it converges

[1]We observe that our probabilistic version of self-stabilization is different from the notion of *randomized self-stabilization* introduced in [14].

to a legitimate configuration, w.h.p. The *convergence time* of a self-stabilizing process is the maximum number of rounds required to reach a legitimate configuration starting from any configuration. This natural notion of (probabilistic) self-stabilization has also been inspired by that in [20] for other distributed processes[2].

Stability has consequences for other important aspects of this process. For instance, if the process is stable, we can get good upper bounds on the *progress* of a ball, namely the number of rounds the ball is selected from its current bin queue, along a sequence of $t \geqslant 1$ rounds. Furthermore, we can eventually bound the *parallel* cover time, i.e., the time required for every ball to visit *all* bins. Self-stabilization has also important consequences when the system is prone to some transient faults [13, 14, 23].

To the best of our knowledge, the repeated balls-into-bins process was first studied in [6] where it is used there as a crucial sub-procedure to optimize the message complexity of a gossip algorithm in the complete graph. The previous analysis in [6, 16] (only) holds for very-short (i.e. logarithmic) periods, while the analysis in [4] considers periods of arbitrary length but it (only) allows to achieve a bound on the maximum load that rapidly increases with time: after t rounds, the maximum load is w.h.p. bounded by $O\left(\sqrt{t}\right)$. By adopting the FIFO strategy at every bin queue, the latter result easily implies that the progress of any ball is w.h.p. $\Omega(\sqrt{t})$. Moreover, it is well known that the cover time for the single-ball process is w.h.p. $\Theta(n \log n)$ (it is in fact equivalent to the *coupon's collector* process [27]). These two facts easily imply an upper bound $O\left(n^2 \log^2 n\right)$ for the parallel cover time of the repeated balls-into-bins process.

Previous results are thus not helpful to establish whether this process is stable (or, even more, self-stabilizing) or not. Moreover, the previous analyses of the maximum load in [4, 6, 16] are far from tight, since they rely on some rough approximations of the studied process via other, much simpler Markov chains: for instance, in [4], the authors consider the process - which clearly dominates the original one - where, at every round, a new ball is inserted in every empty bin. Clearly, that analysis does not exploit the global invariant (a fixed number n of balls) of the original process.

Our Results. We provide a new, tight analysis of the repeated balls-into-bins process that significantly departs from previous ones and show that the system is self-stabilizing. These results are summarized in the following

Theorem 1 *Let c be an arbitrarily-large constant, and let the process start from any legitimate configuration. The maximum load $M^{(t)}$ is $O(\log n)$ for all $t = O(n^c)$, w.h.p. Moreover, starting from any configuration, the system reaches a legitimate configuration within $O(n)$ rounds, w.h.p.*

Our result above strongly improves over the best previous bounds [4, 6, 16] and it is almost tight (since we know that maximum load is $\Omega(\log n/ \log\log n)$ at least during the first rounds [28]). Moreover, the progress of any ball (by adopting the FIFO strategy) over a sequence of $t = \text{poly}(n)$ rounds is $\Omega(t/ \log n)$ w.h.p. and, thus, the parallel cover time is $O(n \log^2 n)$ which is only a $\log n$ factor away from the lower bound arising from the single-ball process.

Besides having *per-se* interest, balls-into-bins processes are used to model and analyze several important randomized protocols in parallel and distributed computing [3, 5, 30]. In particular, the process we study models a natural randomized solution to the problem of *(parallel) resource (or task) assignment* in distributed systems (this problem is also known as *traversal*) [25, 29]. In the basic case, the goal is to assign one resource in mutual exclusion to *all* processors (i.e. nodes) of a distributed system. This is typically described as a *traversal* process performed by a *token* (representing the resource or task) over the network. The process terminates when

[2]On the other hand, we observe that the probabilistic version of self-stabilization adopted here is different from the one introduced in [14], the latter being unsuitable in our context.

the token has visited all nodes of the system. Randomized protocols for this problem [10] are efficient approaches when, for instance, the network is prone to faults/changes and/or when there is no global labeling of the nodes.

A simple randomized protocol is the one based on *random walks* [10, 19, 20]: starting from any node, the token performs a random walk over the network until all nodes are visited, w.h.p. The first round in which all nodes have been visited by the token is called the *cover time* of the random walk [10, 24]. The expected cover time for general graphs is $O(|V| \cdot |E|)$ (see for example [27]).

In distributed systems, we often are in the presence of *several* resources or tasks that must be processed by every node *in parallel*. This naturally leads to consider the parallel version of the basic problem in which n different tokens (resources) are initially distributed over the set of nodes and every token must visit all nodes of the network. Similarly to the basic case, an efficient randomized solution is the one based on (parallel) random walks. In order to visit the nodes, every token performs a random walk under the constraint that every node can process and release at most one token per round. Again, maximum load is a critical complexity measure: for instance, it can determine the required buffer size at every node, bounds on the token progress and, thus, on the parallel cover time.

It is easy to see that, when the graph is complete, the above protocol - based on parallel random walks - is in fact equivalent to the repeated balls-into-bins process analyzed in this paper. For this case, Theorem 1 implies that, every token visits all nodes of the system with at most a logarithmic delay w.r.t. the case of a single token: so, we can derive an upper bound $O(n \log^2 n)$ for the parallel cover time, starting from *any* initial configuration.

We can also consider the adversarial model in which, in some *faulty* rounds, an adversary can re-assign the tokens to the nodes in an arbitrary way. The self-stabilization and the linear convergence time shown in Theorem 1 imply that the $O(n \log^2 n)$ bound on the cover time still holds provided the faulty rounds happen with a frequency not higher than cn, for a sufficiently large constant c.

Related Work

- *Random Walks on Graphs.* As mentioned earlier, the repeated balls-into-bins process was first considered in [4, 6, 16], since it describes the process of performing parallel random walks in the (uniform) gossip model (also known as random phone-call model [11, 21]) when every message can contain at most one token. Maximum load (i.e. node congestion), token delays, mixing and cover times are here the most crucial aspects. We remark that the flavor of these studies is different from ours: indeed, their main goal is to keep maximum load and token delays logarithmic over some *polylogarithmic period.* Their aim is to achieve a fast mixing time for every random walk in the case of good expander graphs. In particular, in [6], a logarithmic bound is shown for the complete graph when $m = O(n/\log n)$ token random walks are performed over a logarithmic time interval. A similar bound is also given for some families of almost-regular random graphs in [16]. Finally, a new analysis is given in [4] for regular graphs yielding the bound $O(\sqrt{t})$.

- *Parallel Computing.* Balls-into-bins processes have been extensively studied in the area of parallel and distributed computing, mainly to address balanced-allocation problems [5, 26, 28], PRAM simulation [22] and hashing [12]. The most studied performance measure is the *maximum load.* In order to optimize the total number of random bin choices used for the allocation, further allocation strategies have been proposed and analyzed (see for instance [8,26,30]). As previously mentioned, our concept of stability is inspired by those studied in [1, 7]. In such works, load balancing algorithms are analyzed in scenarios where new tasks arrive during the run of the

system, and existing jobs are executed by the processors and leave the system.
We remark that, in the above previous works, the goal is different from ours: each ball/task must be allocated to *one, arbitrary* bin/processor. This crucial difference makes such previous analyses of little use to the purpose of our study.

- *Queuing Theory.* To the best of our knowledge, the closest model to our setting in classical queuing theory is the *closed Jackson network* [2]. In this model, time is continuous and each node processes a single token among those in its queue; processing each token takes an exponentially distributed interval of time. As soon as its processing is completed, each token leaves the current node and enters the queue of a neighbor chosen uniformly at random. Notice that, since time is continuous, the process' events are sequential, so that the associated Markov chain is much simpler than the one describing our parallel process. In particular, the stationary distribution of a closed Jackson network can be expressed as a product-form distribution. It is noted in [18] that "[...] virtually all of the models that have been successfully analyzed in classical queuing network theory are models having a so-called product form stationary distribution". Because of the above considerations regarding the difficulty of our process (especially the non-reversibility of its Markov chain), the stationary distribution is instead very likely not to exhibit a product-form distribution, thus laying outside the domain where the techniques of classical queuing theory seem effective. We finally cite the seminal work [9] on *adversarial queing systems*: here, new tokens (having specified source and destination nodes) are inserted in the nodes according to some adversarial strategy and a notion of *edge-congestion* stability is investigated.

2 Self-Stabilization

Overview of the analysis

In the repeated balls-into-bins process, every bin can release at most one ball per round. As a consequence, the random walks performed by the balls delay each other and are thus correlated in a way that can make the bin queues larger than in the independent case. Indeed, intuitively speaking, a large load observed in a bin in some round makes "any" ball more likely to spend several future rounds in that bin, because if the ball ends up in that bin in one of the next few rounds, it will undergo a large delay. This is essentially the major technical issue to cope with.

The previous approach in [4] relies on the fact that, in every round, the expected balance between the number of incoming and outgoing balls is always non-positive for every non-empty bin (notice that the expected number of incoming balls is always at most one). This may suggest viewing the process as a sort of parallel *birth-death* process [24]. Using this approach and with some further arguments, one can (only) get the "standard-deviation" bound $O(\sqrt{t})$ in [4]. Our new analysis proving Theorem 1 proceeds along three main steps.

i) We first show that, after the first round, the aforementioned expected balance is always negative, namely, not larger than $-1/4$. Indeed, the number of empty bins remains at least $n/4$ with (very) high probability, which is extremely useful since a bin can receive tokens only from non-empty bins. This fact is shown to hold starting from *any* configuration and over any period of polynomial length.

ii) In order to exploit the above negative balance to bound the load of the bins, we need some strong concentration bound on the number of balls entering a specific bin u along any period of polynomial size. However, it is easy to see that, for any fixed u, the random variables $\{Z_u^{(t)}\}_{t \geqslant 0}$ counting the number of balls entering bin u are not mutually independent, neither are they negatively associated, so that we cannot apply standard tools to prove concentration. To address this issue, we consider a simpler repeated balls-into-bins process defined as follows.

THE **Tetris** PROCESS. Starting from any configuration with at least $n/4$ empty bins, in each round
- from every non-empty bin we pick one ball and we throw it away, and
- we pick exactly $(3/4)n$ *new balls* and we put each of them independently and u.a.r. in one of the n bins.

Using a coupling argument and our previous upper bound on the number of empty bins, we prove that the maximum number of balls accumulating in a bin in the original process is not larger than the maximum number of balls accumulating in a bin in the `Tetris` process, w.h.p.

iii) The `Tetris` process is simpler than the original one since, at every round, the number of balls assigned to the bins does not depend on the system's state in the previous round. Hence, random variables $\{\hat{Z}_u^{(t)}\}_{t \geqslant 0}$ counting the number of balls arriving at bin u in the `Tetris` process are mutually independent. We can thus apply standard concentration bounds. On the other hand, differently from the approximating process considered in [4], in the `Tetris` process, the negative balance of incoming and outgoing balls proved in Step i) still holds, thus yielding a much smaller bound on the maximum load than that in [4].

In the remainder of this section, we formally describe the above three steps.

Preliminaries and notations

We always use capital letters for random variables, lower case for quantities, and bold for vectors. For each bin $u \in [n]$ let $\mathcal{Q}_u^{(t)}$ be the r.v. indicating the number of balls, i.e. the *load*, in u at round t. We write $\mathbf{Q}^{(t)}$ for the vector of these random variables, i.e., $\mathbf{Q}^{(t)} = \left(\mathcal{Q}_u^{(t)} \,:\, u \in [n]\right)$. We write $\mathbf{q} = (q_1, \ldots, q_n)$ for a *(load) configuration*, i.e., $q_u \in \{0, 1, \ldots, n\}$ for every $u \in [n]$ and $\sum_{u=1}^{n} q_u = n$. In order to enhance readability, in what follows we omit the indication of the round, when it is clear from context, e.g., we write $\mathbf{E}\left[\mathcal{Q}_u \,|\, \mathbf{q}\right]$ for $\mathbf{E}\left[\mathcal{Q}_u^{(t+1)} \mid \mathbf{Q}^{(t)} = \mathbf{q}\right]$.

On the number of empty bins

We next show that the number of *empty* bins is a constant fraction of n for a very large time-window, w.h.p.

Lemma 2 *Let $\mathbf{q} = (q_1, \ldots, q_n)$ be a configuration in a given round and let X be the random variable indicating the number of empty bins in the next round. For any large enough n, it holds that*

$$\mathbf{P}\left(X \leqslant \frac{n}{4}\right) \leqslant e^{-\alpha n},$$

where α is a suitable positive constant.

Proof. Observe that the lemma could be proved by standard concentration arguments if, at every round, *all* balls were thrown independently and uniformly at random. A little care is instead required in our process to consider, at any round, those "congested" bins having load larger than 1. These bins will be surely non-empty at the next round too. So, the number of empty bins at a given round also depends on the number of such congested bins in the previous round. In what follows, we show how two solve this issue by observing a simple but crucial fact.

Let us name $a = a(\mathbf{q})$ the number of empty bins and $b = b(\mathbf{q})$ the number of bins with exactly one token in configuration $\mathbf{q}$. For each bin u of the $a + b$ bins with at most one token, let Y_u be the random variable indicating whether or not bin u is empty in the next round, so that

$$X = \sum_{u=1}^{a+b} Y_u \quad \text{and} \quad \mathbf{P}(Y_u = 1) = \left(1 - \frac{1}{n}\right)^{n-a} \geqslant e^{-\frac{n-a}{n-1}},$$

where in the last inequality we used the fact that $1 - x \geqslant e^{-\frac{x}{1-x}}$. Hence we have that

$$\mathbf{E}[X] \geqslant (a+b)\, e^{-\frac{n-a}{n-1}} \tag{1}$$

The crucial fact is that the number of bins with two or more tokens can be at most as large as the number of empty bins, i.e. $n - (a+b) \leqslant a$. Thus, we can bound the number of empty bins from below[3], $a \geqslant (n-b)/2$, and by using that bound in (1) we get

$$\mathbf{E}[X] \geqslant \frac{n+b}{2}\, e^{-\frac{n+b}{2(n-1)}}$$

Now observe that, for large enough n a positive constant ε exists such that

$$\frac{n+b}{2}\, e^{-\frac{n+b}{2(n-1)}} \geqslant (1+\varepsilon)\frac{n}{4}$$

for every $0 \leqslant b \leqslant n$.

It is not difficult to prove that random variables $Y_1, \ldots, Y_{a+b}$ are *negatively associated* (e.g., see Theorem 13 in [15]). Thus we can apply (see Lemma 7 in [15]) the Chernoff bound (8) with $\delta = \varepsilon/(1+\varepsilon)$ to r.v. X to obtain

$$\mathbf{P}\left(X \leqslant \frac{n}{4}\right) \leqslant \exp\left(-\frac{\varepsilon^2}{4(1+\varepsilon)} n\right)$$

□

From the above lemma it easily follows that, if we look at our process over a time-window $T = T(n)$ of polynomial size, after the first round we always see at least $n/4$ empty bins, w.h.p. More formally, for every $t \in \{1, \ldots, T\}$, let $\mathcal{E}_t$ be the event "The number of empty bins at round t is at least $n/4$". From Lemma 5 and the union bound we get the following lemma (for full-detailed proof see the Appendix).

Lemma 3 *Let $\mathbf{q}_0$ denote the initial configuration, let $T = T(n) = n^c$ for an arbitrarily large constant c. For any large enough n it holds that*

$$\mathbf{P}\left(\bigcap_{t=1}^{T} \mathcal{E}_t \mid \mathbf{Q}^{(0)} = \mathbf{q}_0\right) \geqslant 1 - e^{-\gamma n}$$

where γ is a suitable positive constant.

Coupling with `Tetris`

Using a coupling argument and Lemma 3 we now prove that the maximum load in the original process is stochastically not larger than the maximum load in the `Tetris` process w.h.p.

In what follows we denote by $W^{(t)}$ the *set* of non-empty bins at round t in the original process. Recall that, in the latter, at every round a ball is selected from every non-empty bin

[3] Observe that this argument only works to get a *lower* bound on the number of empty bins and not for an upper bound.

u and it is moved to a bin chosen u.a.r. Accordingly we define, for every round t, the random variables

$$\left\{ X_u^{(t+1)} \,:\, u \in W^{(t)} \right\}, \tag{2}$$

where $X_u^{(t+1)}$ indicates the new position reached in round $t+1$ by the ball selected in round t from bin u. Notice that for every non-empty bin $u \in W^{(t)}$ we have that $\mathbf{P}\left(X_u^{(t+1)} = v \right) = 1/n$ for every bin $v \in [n]$. The random process $\left\{ \mathbf{Q}^{(t)} \,:\, t \in \mathbb{N} \right\}$ is completely defined by random variables X_u^t's, indeed we can write

$$\begin{aligned} \mathcal{Q}_v^{(t+1)} &= \mathcal{Q}_v^{(t)} \dot{-} 1 + \left| \left\{ u \in W^{(t)} \,:\, X_u^{(t+1)} = v \right\} \right| \text{ and} \\ W^{(t+1)} &= \left\{ u \in [n] \,:\, \mathcal{Q}_u^{(t+1)} \geqslant 1 \right\}, \end{aligned}$$

where we used notation $a \dot{-} b = \max\{a-b, 0\}$. For each bin $u \in [n]$, let $\hat{\mathcal{Q}}_u^{(t)}$ be the random variable indicating the number of balls in bin u in round t. We next prove that, over any polynomially-large time window, the maximum load of any bin in our process is stochastically smaller than the maximum number of balls in a bin of the `Tetris` process w.h.p. More formally, we prove the following lemma.

Lemma 4 *Assume we start our process and the* `Tetris` *process from the same initial configuration* $\mathbf{q} = (q_1, \ldots, q_n)$ *such that* $\sum_{u=1}^{n} q_u = n$ *and containing at least* $n/4$ *empty bins. Let* $T = T(n)$ *be an arbitrary round and let* M_T *and* $\hat{M}_T$ *be respectively the random variables indicating the maximum load in our original process and in the* `Tetris` *process, up to round* T. *Formally*

$$\begin{aligned} M_T &= \max\{ \mathcal{Q}_u^{(t)} \,:\, u \in [n],\, t = 1, 2, \ldots, T \} \\ \hat{M}_T &= \max\{ \hat{\mathcal{Q}}_u^{(t)} \,:\, u \in [n],\, t = 1, 2, \ldots, T \} \end{aligned}$$

For every $k \geqslant 0$ *it holds that*

$$\mathbf{P}\left(M_T \geqslant k \right) \leqslant \mathbf{P}\left(\hat{M}_T \geqslant k \right) + T \cdot e^{-\gamma n}$$

for a suitable positive constant γ.

Idea of Proof. We proceed by coupling the `Tetris` process with the original one round by round. Intuitively speaking the coupling proceeds as follows:
- Case (i): *the number of non-empty bins in the original process is* $k \leqslant \frac{3}{4}n$. For each non-empty bin u, let i_u be the ball picked from u. We throw one of the $\frac{3}{4}n$ new balls of the `Tetris` process in the same bin in which i_u ends up. Then, we throw all the remaining $\frac{3}{4}n - k$ balls independently u.a.r.
- Case (ii): *the number of non-empty bins is* $k > \frac{3}{4}n$. We run one round of the `Tetris` process independently from the original one.

By construction, if the number of non-empty bins in the original process is not larger than $\frac{3}{4}n$ at any round, then the `Tetris` process "dominates" the original one, meaning that every bin in the `Tetris` process contains at least as many balls as the corresponding bin in the original one. Since from Lemma 3 we know that the number of non-empty bins in the original process is not larger than $\frac{3}{4}n$ for any time-window of polynomial size w.h.p., we thus have that the `Tetris` process dominates the original process for the whole time window w.h.p. □

Proof. We proceed by coupling the `Tetris` process with the original one as follows. For $t \in \{1, \ldots, T\}$, denote by $B^{(t)}$ the set of new balls in the `Tetris` process at round t (recall that the size of $B^{(t)}$ is $(3/4)n$ for every $t \in \{1, \ldots, T\}$). For any round t and any ball $i \in B^{(t)}$, let $\hat{X}_i^{(t)}$ be the random variable indicating the bin where the ball ends up. Finally, let $\left\{U_i^{(t)} \,:\, t = 1, \ldots, T,\, i \in B^{(t)}\right\}$ be a family of i.i.d. random variables uniform over $[n]$.

At any round $t \in \{1, \ldots, T\}$:

<u>If $|W^{(t-1)}| \leqslant (3/4)n$</u>: Let $B_W^{(t)}$ be an arbitrary subset of $B^{(t)}$ with size exactly $|W^{(t-1)}|$, let $f^{(t)} \,:\, B_W^{(t)} \to W^{(t-1)}$ be an arbitrary bijection and set

$$\hat{X}_i^{(t)} = \begin{cases} X_i^{(t)} & \text{if } i \in B_W^{(t)} \\ U_i^{(t)} & \text{if } i \in B^{(t)} \setminus B_W^{(t)} \end{cases} \tag{3}$$

<u>If $|W^{(t-1)}| > (3/4)n$</u>: Set $\hat{X}_i^{(t)} = U_i^{(t)}$ for all $i \in B^{(t)}$.

By construction we have that random variables

$$\left\{\hat{X}_i^{(t)} \,:\, t \in \{1, 2, \ldots, T\},\, i \in B^{(t)}\right\}$$

are mutually independent and uniformly distributed over $[n]$. Moreover, in the joint probability space for any k we have that

$$\mathbf{P}\left(M_T \geqslant k\right) = \mathbf{P}\left(M_T \geqslant k,\, \hat{M}_T \geqslant M_t\right) + \mathbf{P}\left(M_T \geqslant k,\, \hat{M}_T < M_T\right) \leqslant$$
$$\leqslant \mathbf{P}\left(\hat{M}_t \geqslant k\right) + \mathbf{P}\left(\hat{M}_T < M_T\right)$$

Finally, let $\mathcal{E}_T$ be the event "There are at least $n/4$ empty bins at all rounds $t \in \{1, \ldots, T\}$" and observe that, from the coupling we have defined, the event $\mathcal{E}_T$ implies event "$\hat{M}_T \geqslant M_T$". Hence $\mathbf{P}\left(\hat{M}_T < M_T\right) \leqslant \mathbf{P}\left(\overline{\mathcal{E}}\right)$ and the thesis follows from Lemma 3. □

In the `Tetris` process, the random variables indicating the number of balls ending up in a bin in different rounds are i.i.d. binomial. This fact is extremely useful to give upper bounds on the load of the bins, as we do in the next simple lemma, that will be used to prove self-stabilization of the original process.

Lemma 5 *From any initial configuration, in the `Tetris` process every bin will be empty at least once within $5n$ rounds w.h.p.*

Proof. Let $u \in [n]$ be a bin with $k \leqslant n$ balls in the initial configuration. For $t \in \{1, \ldots, 5n\}$ let Y_t be the random variable indicating the number of new balls ending up in bin u at round t. Notice that in the `Tetris` process $Y_1, \ldots, Y_{5n}$ are i.i.d. $Bin\left((3/4)n,\, 1/n\right)$ hence $\mathbf{E}\left[Y_1 + \cdots + Y_{5n}\right] = (15/4)n$ and by applying Chernoff bound (9) with $\delta = 1/15$ we get

$$\mathbf{P}\left(Y_1 + \cdots + Y_{5n} \geqslant 4n\right) \leqslant e^{-\alpha n}$$

where $\alpha = 1/(665)$.

Now let $\mathcal{E}_u$ be the event *"Bin u will be non-empty for all the $5n$ rounds"*. Since when a bin is non-empty it looses a ball at every round, event $\mathcal{E}_u$ implies, in particular, that

$$k - 5n + Y_1 + \cdots + Y_{5n} \geqslant 0$$

That is $Y_1 + \cdots + Y_{5n} \geqslant 5n - k \geqslant 4n$. Thus

$$\mathbf{P}\left(\mathcal{E}_u\right) \leqslant \mathbf{P}\left(Y_1 + \cdots + Y_{5n} \geqslant 4n\right) \leqslant e^{-\alpha n}$$

The thesis follows from the union bound over all bins $u \in [n]$. □

On the maximum load in the `Tetris` process

We next focus on the maximum load that can be observed in the `Tetris` process at any given bin within a finite interval of time.

Recall the definition of X_i^t in (3) and let $I_i^t(u) = [X_i^t = u]$. Consider an interval $[\tau_1, \tau_2]$. We denote by $Z_u^{[\tau_1,\tau_2]}$ the overall number of balls that enter bin u during $[\tau_1, \tau_2]$, namely:

$$Z_u^{[\tau_1,\tau_2]} = \sum_{t=\tau_1}^{\tau_2} \sum_{i \in B^{(t)}} I_i^t(u)$$

By linearity of expectation we get the following lemma (see Appendix for details).

Lemma 6 *For any $\tau > 0$ and $\Delta \in \{0, \ldots, \tau - 1\}$, in the `Tetris` process it holds that*

$$\mathbf{E}\left[Z_u^{[\tau-\Delta,\tau]}\right] = \frac{3}{4}(\Delta + 1) \tag{4}$$

Considered a bin u and a time t, we denote by $T_u(t)$ the last time, prior to t, such that u was empty, namely

$$T_u(t) = \max\{\tau \mid \tau \leqslant t, \hat{\mathcal{Q}}_u^{(\tau)} = 0\}$$

We set $T_u(t) = 0$ when the bin was never empty in the interval $[1, t]$. We next use the fact that, if the load at some bin u is sufficiently high at the end of a given round t, there exists a contiguous time interval ending at t, during which a number of balls significantly deviating from the expectation in (4). This simple fact is formalized in the next lemma.

Lemma 7 *Consider a generic bin u that has been empty at some time τ_1. For any $\alpha > 0$ and $\tau_2 > \tau_1$, it holds*

$$\mathbf{P}\left(\hat{\mathcal{Q}}_u^{(\tau_2)} > \alpha\right) \leqslant \sum_{\Delta=0}^{\tau_2-\tau_1} \mathbf{P}\left(Z_u^{[\tau_2-\Delta,\tau_2]} > \Delta + \alpha\right) \tag{5}$$

Proof. From the definition of the `Tetris` process, it is easy to see that the event "$\hat{\mathcal{Q}}_u^{(\tau_2)} > \alpha$ and $T_u(\tau_2) = \tau_2 - \Delta - 1$" implies the arrival of $Z_u^{[\tau_2-\Delta,\tau_2]} > \Delta + \alpha$ balls in the interval $[\tau_2 - \Delta, \tau_2]$, that is

$$(\hat{\mathcal{Q}}_u^{(\tau_2)} \geqslant \alpha \bigwedge T_u(\tau_2) = \tau_2 - \Delta - 1) \Rightarrow (Z_u^{[\tau_2-\Delta,\tau_2]} \geqslant \Delta + \alpha) \tag{6}$$

As a consequence, for every t and $\alpha > 0$ we have:

$$\mathbf{P}\left(\hat{\mathcal{Q}}_u^{(\tau_2)} > \alpha\right) = \sum_{t=\tau_1}^{\tau_2} \mathbf{P}\left(\hat{\mathcal{Q}}_u^{(\tau_2)} > \alpha \bigwedge T(\tau_2) = t\right) \leqslant \sum_{\Delta=0}^{\tau_2-\tau_1} \mathbf{P}\left(Z_u^{[\tau_2-\Delta,\tau_2]} > \Delta + \alpha\right),$$

where the first equality follows from the fact that for $t \neq \hat{t}$ the events "$\hat{\mathcal{Q}}_u^{(\tau_2)} > \alpha \bigwedge T(\tau_2) = t$" and "$\hat{\mathcal{Q}}_u^{(\tau_2)} > \alpha \bigwedge T(\tau_2) = \hat{t}$" are disjoint, and the last inequality follows from (6). □

Thanks to Lemma 7, we are able to prove the following key property of the load observed on any bin in the `Tetris` process.

Lemma 8 *Consider a generic bin u in the `Tetris` process, and let τ_1 be a round in which u was empty, namely $\hat{\mathcal{Q}}_u^{(\tau_1)} = 0$. Let τ_2 be any round such that $\tau_2 > \tau_1$. For any constant $\beta > 0$, it holds that*

$$\mathbf{P}\left(\exists t \in [\tau_1, \tau_2] : \hat{\mathcal{Q}}_u^{(t)} > \frac{192}{5}\beta \cdot \log n\right) \leqslant \frac{(\tau_2 - \tau_1 + 1)^2}{n^\beta} \tag{7}$$

Proof. Since $Z_u^{[t-\Delta,t]}$ is a sum of i.i.d. r.v.s (namely the X_i^ts), for any $t \in [\tau_1, \tau_2]$ and $\Delta \in [0, t-\tau_1]$ we can apply the Chernoff bound (9) with $\delta = \frac{1}{4}$ and $\mu_H^{(\Delta)} = \max\{\frac{3}{4}(\Delta+1), 48\beta \cdot \log n\}$ (from Lemma (6), note that $\mu_H^{(\Delta)} \geqslant \mathbf{E}\left[Z_u^{[\tau_2-\Delta,\tau_2]}\right]$). Thus, from Lemma 7 with $\alpha = \frac{192}{5}\beta \cdot \log n$ we get

$$\mathbf{P}\left(\hat{\mathcal{Q}}_u^{(t)} > \frac{192}{5}\beta \cdot \log n\right) \leqslant \sum_{\Delta=0}^{t-\tau_1} \mathbf{P}\left(Z_u^{[t-\Delta,t]} > \Delta + \frac{192}{5}\beta \cdot \log n\right) \leqslant$$
$$\leqslant \sum_{\Delta=0}^{t-\tau_1} \mathbf{P}\left(Z_u^{[t-\Delta,t]} > \mu_H^{(\Delta)}(1+\delta)\right) \leqslant \sum_{\Delta=0}^{t-\tau_1} \exp\left(-\frac{\delta^2}{3}\mu_H^{(\Delta)}\right) \leqslant (\tau_2 - \tau_1 + 1)\exp(-\beta \log n)$$

Finally, the thesis follows from the above bound on the events "$\hat{\mathcal{Q}}_u^{(t)} > \frac{192}{5}\beta \cdot \log n$" and the union bound on their union for $t \in [\tau_1, \tau_2]$. Using Lemma 8 and the union bound on all bins, we easily get the following bound on the maximum load in the `Tetris` process.

Theorem 9 *Let c be an arbitrarily-large constant, and let the `Tetris` process start from any legitimate configuration. The maximum load $\hat{M}^{(t)}$ is $O(\log n)$ for all $t = O(n^c)$, w.h.p.*

Back to the original process: Proof of Theorem 1

From a standard balls-into-bins argument (see [27]), starting from any legitimate configuration, after one round the process still lies in a legitimate configuration w.h.p. and, thanks to Lemma 2, there are at least $n/4$ empty bins w.h.p. From Lemma 4 with $T = O(n^c)$ we thus have that the maximum load of the process is not larger than the maximum load of the `Tetris` process in all rounds $1, ..., T$ w.h.p. Finally, the upper bound on the maximum load of the `Tetris` process in Theorem 9 completes the proof of the first statement in Theorem 1.

As for self-stabilization, given an arbitrary initial configuration, from Lemma 5 it follows that within $O(n)$ rounds all bins have been empty at least once w.h.p. When a bin becomes empty, Lemma 8 implies that its load will stay $O(\log n)$ for a polynomial number of rounds. Hence, within $O(n)$ rounds, the system will reach a legitimate configuration w.h.p. □

3 Parallel Resource Assignment

As mentioned in the introduction, the repeated balls-into-bins process can also be seen as running parallel random walks of n distinct tokens (i.e. balls), each of them starting from a node (i.e. bins) of the complete graph of size n. This is a randomized protocol for the parallel allocation problem where tokens represent different resources/tasks that must be assigned to all nodes in mutual exclusion [10]. In this scenario, a critical complexity measure is the (global) cover time, i.e., the time required by any token to visit all nodes.

It is important to observe that our analysis on self-stabilization works for anonymous tokens and nodes and, hence, for any particular queuing strategy. In order to bound the delay of any token, we can consider the FIFO strategy to select tokens from every bin queue. According to this strategy, we have that every token in every bin never waits more than a number of rounds larger than the maximum load. Hence, Theorem 1 implies that, starting from any initial token assignment and for a period of polynomial length, every token will stay in every bin queue for at most a logarithmic number of rounds, w.h.p. We also known that the cover time of the single random-walk process is w.h.p. $O(n \log n)$ [27]. Combining the above two facts, we easily get the following result.

Corollary 10 *The random-walk protocol for the Parallel Resource Assignment problem on the clique has cover time $O(n \log^2 n)$, w.h.p.*

Adversarial model

The self-stabilization property shown in Theorem 1 makes the random walk protocol robust to some transient faults. We can consider an adversarial model in which, in some *faulty* rounds, an adversary can reassign the tokens to the nodes in an arbitrary way. Then, the linear convergence time shown in Theorem 1 implies that the $O(n \log^2 n)$ bound on the cover time still holds provided the faulty rounds happen with a frequency not higher than γn, for any constant $\gamma \geqslant 6$. Indeed, thanks to Lemma 5, the action of an adversary manipulating the system configuration once every γn rounds can affect only the successive $5n$ rounds, while our analysis in the non-adversarial model does hold for the remaining $(\gamma - 5)n$ rounds. It follows that the overall slowdown on the cover time produced by such an adversary is at most a constant factor on the previous $O(n \log^2 n)$ upper bound, w.h.p.

4 Conclusions and Open Questions

We have shown that repeated balls-into-bin is self-stabilizing when the number m of balls is equal to the number n of bins. This clearly holds when $m < n$ as well. An interesting open question is whether this self-stabilization property also holds for a larger number of balls, i.e., for any $m = O(n \log n)$. We believe that an approach based on a lower bound on the number of empty bins might still work. Computer experiments on increasing system sizes (up to $n \sim 10^5$) seem to open a chance for this result: the number of empty bins are compatible with a linear function, even though the standard deviation in our experiments turns out to be relatively large.

A more general interesting question is the study of this process over other classes of graphs. This line of research is also motivated by several recent applications of parallel random walks in the (uniform) gossip model [6, 10, 16, 17]. As mentioned in the introduction, the previous analysis of this process in regular graphs [4] yields a bound on the maximum load $O(\sqrt{t})$ after t rounds. As we proved here for the complete graph, we believe that the previous bound is far from tight even in regular graphs and it leads to very rough bounds on the parallel cover time. We conjecture that the maximum load remains logarithmic for a long period in any *regular* graph. A possible reason for this important phenomenon (if true) might be the fact that in regular graphs the expected difference between (token) arrivals and departures is always *non-positive* in every node. As in our analysis on the complete graph, this fact is not enough but, if it could be combined with a suitable bound on the number of empty bins, then it could lead to the right way for proving our conjecture. However, in non-complete graphs, there is a further technical issue: in order to apply any argument on the empty bins, we also need to prove that such empty bins keep *well spread* over (almost) all neighborhoods of the regular graph for a long period. We think this technical issue is far from easy even in simple topologies such as rings.

Acknowledgments

We would like to thank Riccardo Silvestri for helpful discussions and important hints.

References

[1] A. Anagnostopoulos, A. Kirsch, and E. Upfal. Load balancing in arbitrary network topologies with stochastic adversarial input. *SIAM Journal on Computing*, 34(3):616–639, 2005.

[2] S. Asmussen. *Applied probability and queues.* Springer, 2003.

[3] Y. Azar, A. Z. Broder, A. R. Karlin, and E. Upfal. Balanced allocations. *SIAM journal on computing*, 29(1):180–200, 1999.

[4] L. Becchetti, A. Clementi, E. Natale, F. Pasquale, and R. Silvestri. Plurality consensus in the gossip model. In *Proceedings of the 25th ACM-SIAM Symposium on Discrete Algorithms (SODA)*, 2015.

[5] P. Berenbrink, A. Czumaj, A. Steger, and B. Vöcking. Balanced allocations: The heavily loaded case. *SIAM Journal on Computing*, 35(6):1350–1385, 2006.

[6] P. Berenbrink, J. Czyzowicz, R. Elsässer, and L. Gasieniec. Efficient information exchange in the random phone-call model. In *Proceedings of the 37th International Colloquium on Automata, Languages, and Programming (ICALP)*, 2010.

[7] P. Berenbrink, T. Friedetzky, and L. A. Goldberg. The natural work-stealing algorithm is stable. *SIAM Journal on Computing*, 32(5):1260–1279, 2003.

[8] P. Berenbrink, K. Khodamoradi, T. Sauerwald, and A. Stauffer. Balls-into-bins with nearly optimal load distribution. In *Proceedings of the 25th ACM Symposium on Parallelism in Algorithms and Architectures (SPAA)*, 2013.

[9] A. Borodin, J. Kleinberg, P. Raghavan, M. Sudan, and D. P. Williamson. Adversarial queuing theory. *Journal of the ACM*, 48(1):13–38, 2001.

[10] C. Cooper. Random walks, interacting particles, dynamic networks: Randomness can be helpful. In *Proceedings of the 37th International Colloquium on Structural Information and Communication Complexity (SIROCCO)*, 2011.

[11] A. Demers, D. Greene, C. Hauser, W. Irish, J. Larson, S. Shenker, H. Sturgis, D. Swinehart, and D. Terry. Epidemic algorithms for replicated database maintenance. In *Proceedings of the 6th ACM Symposium on Principles of Distributed Computing (PODC)*, 1987.

[12] M. Dietzfelbinger, A. Goerdt, M. Mitzenmacher, A. Montanari, R. Pagh, and M. Rink. Tight thresholds for cuckoo hashing via xorsat. In *Proceedings of the 37th International Colloquium on Automata, Languages, and Programming (ICALP)*, 2010.

[13] E. W. Dijkstra. Self-stabilizing systems in spite of distributed control. *Communications of the ACM*, 17(11):643–644, 1974.

[14] S. Dolev. *Self-stabilization.* MIT press, 2000.

[15] D. P. Dubhashi and D. Ranjan. Balls and bins: A study in negative dependence. *Random Structures & Algorithms*, 13(2):99–124, 1998.

[16] R. Elsässer and D. Kaaser. On the influence of graph density on randomized gossiping. *Proceedings of the 29th IEEE International Parallel & Distributed Processing Symposium (IPDPS)*, 2015.

[17] B. Haeupler, G. Pandurangan, D. Peleg, R. Rajaraman, and Z. Sun. Discovery through gossip. In *Proceedings of the 24th ACM Symposium on Parallelism in Algorithms and Architectures (SPAA)*, 2012.

[18] J. M. Harrison and R. Williams. Brownian models of feedforward queueing networks: Quasireversibility and product form solutions. *The Annals of Applied Probability*, pages 263–293, 1992.

[19] S. Ikeda, I. Kubo, N. Okumoto, and M. Yamashita. Fair circulation of a token. *IEEE Transactions on Parallel and Distributed Systems*, 13(4):367–372, 2002.

[20] A. Israeli and M. Jalfon. Token management schemes and random walks yield self-stabilizing mutual exclusion. In *Proceedings of the 9th annual ACM Symposium on Principles of Distributed Computing (PODC)*, 1990.

[21] R. Karp, C. Schindelhauer, S. Shenker, and B. Vocking. Randomized rumor spreading. In *Proceedings of the 41th Annual IEEE Symposium on Foundations of Computer Science (FOCS)*, 2000.

[22] R. M. Karp, M. Luby, and F. M. auf der Heide. Efficient pram simulation on a distributed memory machine. *Algorithmica*, 16(4-5):517–542, 1996.

[23] L. Lamport. Solved problems, unsolved problems and non-problems in concurrency. *ACM SIGOPS Operating Systems Review*, 19(4):34–44, 1985.

[24] D. A. Levin, Y. Peres, and E. L. Wilmer. *Markov chains and mixing times.* American Mathematical Society, 2009.

[25] N. A. Lynch. *Distributed algorithms.* Morgan Kaufmann, 1996.

[26] M. Mitzenmacher, B. Prabhakar, and D. Shah. Load balancing with memory. In *Proceedings of the 43th Annual IEEE Symposium on Foundations of Computer Science (FOCS)*, 2002.

[27] M. Mitzenmacher and E. Upfal. *Probability and computing: Randomized algorithms and probabilistic analysis.* Cambridge University Press, 2005.

[28] M. Raab and A. Steger. "balls into bins"—a simple and tight analysis. In *Proceedings of the 2nd International Workshop on Randomization and Approximation Techniques in Computer Science (RANDOM)*, 1998.

[29] N. Santoro. *Design and analysis of distributed algorithms.* John Wiley & Sons, 2006.

[30] B. Vöcking. How asymmetry helps load balancing. *Journal of the ACM*, 50(4):568–589, 2003.

Appendix

A Useful inequalities

Lemma 11 (Chernoff bound) *Let $\{X_t : t \in [n]\}$ be a family of independent binary random variables. Let $X = \sum_{t=1}^{n} X_t$ and let $\mu_L \leqslant \mathbf{E}[X] \leqslant \mu_H$. For every $\delta \in (0,1)$ and $\beta > 0$ it holds that*

$$\mathbf{P}(X \leqslant (1-\delta)\mu_L) \leqslant \exp\left(-\frac{\delta^2}{2}\mu_L\right) \tag{8}$$

$$\mathbf{P}(X \geqslant (1+\delta)\mu_H) \leqslant \exp\left(-\frac{\delta^2}{3}\mu_H\right) \tag{9}$$

B Negative association

Definition 12 (Negative association) *Random variables $X_1, \ldots, X_n$ are* negatively associated *if, for every pair of disjoint subsets $I, J \subseteq [n]$, it holds that*

$$\mathbf{E}[f(X_i,\, i \in I) \cdot g(X_j,\, j \in J)] \leqslant \mathbf{E}[f(X_i,\, i \in I)] \cdot \mathbf{E}[g(X_j,\, j \in J)]$$

for all pairs of functions $f : \mathbb{R}^{|I|} \to \mathbb{R}$ and $g : \mathbb{R}^{|J|} \to \mathbb{R}$ that are both non-decreasing or both non-increasing.

Now we give a simple counterexample showing that, in our balls-into-bins process, the random variables counting the number of balls arriving in a given bin in different rounds cannot be negatively associated.

Consider our random process with $n = 2$ and let X_1 and X_2 be the random variables indicating the number of tokens arriving at the first bin in rounds 1 and 2, respectively. Let $f \equiv g$ be the non-increasing function

$$f(x) = \begin{cases} 1 & \text{if } x = 0 \\ 0 & \text{if } x > 0 \end{cases}$$

If X_1 and X_2 were negatively associated, we thus would have that $\mathbf{P}(X_1 = 0,\, X_2 = 0) \leqslant \mathbf{P}(X_1 = 0)\,\mathbf{P}(X_2 = 0)$. However, by direct calculation it is easy to compute that

$$\mathbf{P}(X_1 = 0,\, X_2 = 0) = 1/8$$

because, in order for "$X_1 = 0,\, X_2 = 0$" to happen, at the first round both balls have to end up in the second bin (this happens with probability 1/4) and at the second round the ball chosen in the second bin has to stay there (this happens with probability 1/2). But we have that $\mathbf{P}(X_1 = 0) = 1/4$ and by conditioning on all the three possible configurations at round 1 we have $\mathbf{P}(X_2 = 0) = 3/8$. Thus

$$\frac{1}{8} = \mathbf{P}(X_1 = 0,\, X_2 = 0) > \mathbf{P}(X_1 = 0)\,\mathbf{P}(X_2 = 0) = \frac{1}{4} \cdot \frac{3}{8}$$

In general, intuitively speaking it seems that event "$X_t = 0$" makes more likely the event that there are a lot of empty bins in the system, which in turn makes more likely event "$X_{t+1} = 0$" that the bin will receive no tokens at round $t + 1$ as well.

C Omitted proofs

Proof of Lemma 3

By using the union bound we have that

$$\mathbf{P}\left(\bigcap_{t=1}^{T} \mathcal{E}_t \mid \mathbf{Q}^{(0)} = \mathbf{q}_0\right) = 1 - \mathbf{P}\left(\bigcup_{t=1}^{T} \overline{\mathcal{E}_t} \mid \mathbf{Q}^{(0)} = \mathbf{q}_0\right) \geqslant \ 1 - \textstyle\sum_{t=1}^{T} \mathbf{P}\left(\overline{\mathcal{E}_t} \mid \mathbf{Q}^{(0)} = \mathbf{q}_0\right)$$

By conditioning on the configuration at round $t-1$, from the Markov property and Lemma 2 it then follows that

$$\mathbf{P}\left(\overline{\mathcal{E}_t} \mid \mathbf{Q}^{(0)} = \mathbf{q}_0\right) = \sum_{\mathbf{q}} \mathbf{P}\left(\overline{\mathcal{E}_t} \mid \mathbf{Q}^{(t-1)} = \mathbf{q}\right) \mathbf{P}\left(\mathbf{Q}^{(t-1)} = \mathbf{q} \mid \mathbf{Q}^{(0)} = \mathbf{q}_0\right) \leqslant e^{-\alpha n}$$

Hence,

$$\mathbf{P}\left(\bigcap_{t=1}^{T} \mathcal{E}_t \mid \mathbf{Q}^{(0)} = \mathbf{q}_0\right) \geqslant 1 - Te^{-\alpha n} \geqslant 1 - e^{-\gamma n}$$

for a suitable positive constant γ. □

Proof of Lemma 6

Recall that in the `Tetris` process $|B^{(t)}| = (3/4)n$ for every $t > 1$. Thus, from the definition of $Z_u^{[\tau-\Delta,\tau]}$ we get

$$\mathbf{E}\left[Z_u^{[\tau-\Delta,\tau]}\right] = \sum_{t=\tau-\Delta}^{\tau} \sum_{i \in B^{(t)}} \mathbf{E}\left[I_i^t(u)\right] = \sum_{t=\tau-\Delta}^{\tau} \sum_{i \in B^{(t)}} \frac{1}{n} = \frac{3}{4}(\Delta + 1)$$

□