Balls and Bins with Structure: Balanced Allocations on Hypergraphs

P. Brighten Godfrey*

Abstract

In the standard balls-and-bins model of balanced allocations, m balls are placed sequentially into n bins. Each ball chooses d uniform-random bins and is placed in the least loaded bin. It is well known that when $d = \log^{\Theta(1)} n$, after placing $m = n$ balls, the maximum load (number of balls in a bin) is $\Theta(1)$ w.h.p.

In this paper we show that as long as $d = \Omega(\log n)$, independent random choices are not necessary to achieve a constant load balance: these choices may be structured in a very general way. Specifically, we allow each ball i to have an associated random set of bins B_i. We require that $|B_i| = \Omega(\log n)$ and that bins are included in B_i with approximately the same probability; but the distributions of the B_is are otherwise arbitrary, so that there may be correlations in the choice of bins. We show that this model captures structure important to two applications, nearby server selection and load balance in distributed hash tables.

1 Introduction

In the standard balls-and-bins model of randomized load balancing, m balls are placed sequentially into n bins. Each ball probes the current load of d uniform-random bins and is placed in the least loaded bin. The *maximum load* is the maximum number of balls in any bin at the end of the process. It is well known that when $d = 1$, this procedure results in a maximum load of $\Theta\left(\frac{\log n}{\log \log n}\right)$ with high probability, and Azar et al [1] showed that if $d > 1$, the maximum load is $\frac{\log \log n}{\log d} + O(1)$ w.h.p. (see also [8]). In particular, when $d = \log^{\Theta(1)} n$, the maximum load is $O(1)$ w.h.p.

Many subsequent works have studied placement algorithms which do not probe independent and uniform-random bins. For example, Vöcking [12] improved the maximum load to $\frac{\log \log n}{d \ln \phi_d} + O(1)$ where $1 \leq \phi_d \leq 2$ by using a placement algorithm in which the bins are split into d groups, each ball probes a random bin from each group, and ties are broken asymmetrically. This yields constant maximum load with $d = \Theta(\log \log n)$. Byers et al [2] showed that if the probed bins are picked independently as in the original model but with somewhat nonuniform probabilities, the maximum load can still be bounded by $\frac{\log \log n}{\log d} + O(1)$. Wieder [13] characterized the nonuniform case when $m \gg n$.

None of these placement algorithms addresses the fact that for some applications, probing certain sets of d bins is more costly than other sets. Consider, for example, a client arriving in a random location that wishes to connect to a lightly-loaded server. Probing the d *closest* servers (i.e., bins) is likely to be much more desirable than probing d *random* servers, both because the probes themselves will be cheaper, and because the client will ultimately receive service from a much closer server. To give applications such flexibility, Kenthapadi and Panigrahy [5] studied more general bin-selection processes for the case $d = 2$ by allowing each ball to choose a random pair of bins from only a subset of the possible pairs. This can be represented by a graph G whose nodes are bins, and whose edges are the allowable pairs. The standard result stated within this model is that the maximum load is $\frac{\log \log n}{\log 2} + O(1)$ when G is the complete graph. But [5] showed that G need not be complete: if G is almost regular with degree n^ε, the maximum load is $\log \log n + O(1/\varepsilon)$, which for $\varepsilon = \Theta(1/\log \log n)$ is within a constant factor of the maximum load in the standard model. Following the client-server example, this would allow a client to consider pairs of servers among the $n^{\Theta(1/\log \log n)}$ closest servers while still obtaining a maximum load of $O(\log \log n)$.

Model and Main Results. The present paper studies structured choices for other values of d while allowing even more flexible choice of bins. Our model, which we will relate to that of [5] below, is simply to allow each ball i to probe an associated random set of bins B_i, whose distribution and size (i.e., d) may differ for each ball. Note that this permits arbitrary correlations between the probed bins in each B_i. What restrictions on the distributions of the B_is are sufficient to guarantee a good maximum load?

Somewhat surprisingly, when $|B_i| = \Omega(\log n)$, we

*CS Division, UC Berkeley. Supported in part by a National Science Foundation Graduate Research Fellowship. Email: pbg@cs.berkeley.edu.

need only require that each B_i satisfies a *balance* property. In the simple case that d is the same for all balls, B_i is balanced when for each bin j, $\Pr[j \in B_i]$ is within a constant factor of the "fair" probability, $\frac{d}{n}$. Our main theorem is that, under these conditions and for a constant α to be specified, placing $m = \alpha n$ balls in n bins results in a maximum load of 1 with high probability, and placing $m = n$ balls results in a maximum load of $\lceil 1/\alpha \rceil$. Although the conditions are quite intuitive, the difficulty in the proof is to deal with the dependencies between each ball's choices. We handle this by showing that the distribution of balls in bins is approximately dominated by a process of placing slightly more than m balls into uniform-random least-loaded bins.

We also note a lower bound, that for any $1 \leq d \leq \Theta\left(\frac{\log n}{\log \log n}\right)$, there exist balanced B_is for which the maximum load is $\frac{1}{d} \cdot \frac{\ln n}{\ln \ln n} \cdot (1 + o(1))$ w.h.p. Thus, the power of two choices result, that the maximum load decreases to $\Theta(\log \log n)$ when $d = 2$, does not hold in our model. Intuitively, this is because correlations in the bin choices can be such that a ball which picks a "hotspot" bin shares the same d alternative choices as all the other $\approx \frac{\ln n}{\ln \ln n}$ balls that picked the same hotspot. Our main theorem shows that $d = \Omega(\log n)$ alternatives are sufficient to spread the load of hotspots, regardless of the pattern of bin choices.

Relation to Balanced Allocations on Graphs. The present paper's setting can be related to the model of [5] by generalizing balanced allocations on a graph to balanced allocations on a hypergraph G. Each hyperedge of G represents an allowable set of d bins, and as before each ball probes the bins in a uniform-random hyperedge. A special case of our theorem is that when $d = \Theta(\log n)$, the maximum load is $\Theta(1)$ as long as G is almost regular (that is, the minimum and maximum node degrees differ by at most a constant factor). This permits as few as one hyperedge incident to each node in the hypergraph, which translates to a minimum of $\Theta(\log n)$ neighbors per node. Comparing this with the $n^{\Theta(1/\log \log n)}$ minimal degree for the case $d = 2$ shows that increasing the number of probes to $d = \Theta(\log n)$ substantially decreases the connectivity necessary to obtain a maximum load within a constant factor of that in the standard balls-and-bins model. Continuing the earlier client-server example, which will be treated more formally later, a client arriving in a random location can restrict its attention to the $\Theta(\log n)$ closest servers rather than the $n^{\Theta(1/\log \log n)}$ required when $d = 2$.

Applications. Finally, we describe two applications of our main result. The first, nearby server selection, fleshes out the client-server example above. The second application involves load-balanced file placement in distributed hash tables (DHTs). We describe how a scheme of Byers et al [2] can, in an amenable DHT structure, be modified to obtain a better balance while sending the same total number of messages. By structuring choices in file placement to match the DHT's topology, each choice is less costly than a random choice. These examples suggest that our model of correlated bin choice captures structure important to applications.

Our main theorem and the lower bound are proved in Sections 2 and 3, respectively. We discuss the two applications in Section 4, and conclude in Section 5.

2 Main Theorem

We are given n bins into which will be placed $m = \alpha n$ balls. For each ball i we are given a distribution $\mathcal{B}_i$ over sets of bins. Each ball is placed according to the following algorithm which we will call Algorithm $\mathbf{A}_i$:

1. Let B_i be a random set of bins distributed as $\mathcal{B}_i$ (chosen independently of other selections).
2. Place ball i in a uniform-random bin from among those bins in B_i with the fewest number of balls.

DEFINITION 2.1. *A random set of bins B is β-*balanced *if, for all bins j,*

$$\frac{1}{\beta n} \leq \Pr[j \in B] \cdot E\left[\frac{1}{|B|} \mid j \in B\right] \leq \frac{\beta}{n}.$$

THEOREM 2.1. *Let $\varepsilon > 0$, $\delta \in (0,1)$, and suppose that for each ball i, B_i is β-balanced and $|B_i| \geq 26 \cdot \frac{(1+\varepsilon)^2}{\varepsilon^2 \delta} \cdot \log n$. Let $\beta' = (1 + \varepsilon + o(1)) \cdot \beta$ and $\alpha = (1-\delta)/\left\lceil 1 - \frac{\ln \beta'}{\ln(1-(\beta'-1)/(\beta'^2-1))} \right\rceil$. Then with probability $\geq 1 - O(n^{-1})$, the maximum load is 1 after placing $m = \alpha n$ balls, and the maximum load is $\lceil 1/\alpha \rceil$ after placing $m = n$ balls.*

2.1 Proof of Theorem. For convenience, unless otherwise stated, the proof and associated lemmas assume that $m = \alpha n$ (the case $m = n$ is an easy final step).We analyze Algorithm **A** by showing that it is dominated by Algorithm **B**: for each ball **A** places with structured choices, **B** will place a ball into each of k uniform-random empty bins, where the constant k will be selected later (in the proof of Lemma 2.2) such that $mk \leq n$. We first state several definitions and lemmas, which will be proved in later subsections, and then prove the theorem. The bulk of the technical material actually appears in the proof of Lemma 2.2.

DEFINITION 2.2. *A* coupling *of two random allocations of balls in bins, D and D', is a pair of functions f, f' :*

$[0,1] \to \{0,\ldots,m\}^n$ *such that, if* R *is a random variable uniform on* $[0,1]$*, then* $f(R)$ *and* $f'(R)$ *are distributed as* D *and* D'*, respectively.*

DEFINITION 2.3. *Given two random allocations* D, D' *of balls in bins,* D is ε-dominated by D'*, written* $D \preceq_\varepsilon D'$*, when there is a coupling* (f, f') *of* D *and* D' *such that* $\Pr[f(R)_j \leq f'(R)_j \, \forall j] \geq 1-\varepsilon$*, where* $f(\cdot)_j$ *denotes the number of balls in bin* j *in the allocation* $f(\cdot)$*.*

Let $\mathbf{X} \circ D$ denote the random allocation that results from application of an algorithm $\mathbf{X}$ to the random allocation D of balls in bins. Define $P1(i)$ and $P2(i)$ as the random allocations that result from applying algorithms $\mathbf{A}$ and $\mathbf{B}$, respectively, i times, beginning with the allocation Z of zero balls in the bins. That is,

$$P1(i) = \begin{cases} Z & \text{if } i = 0 \\ \mathbf{A}_i \circ P1(i-1) & \text{if } i > 0 \end{cases}$$

$$P2(i) = \begin{cases} Z & \text{if } i = 0 \\ \mathbf{B} \circ P2(i-1) & \text{if } i > 0. \end{cases}$$

LEMMA 2.1. *If* $D \preceq_\varepsilon E$ *then* $\mathbf{A}_i \circ D \preceq_\varepsilon \mathbf{A}_i \circ E \; \forall i$*.*

LEMMA 2.2. $\mathbf{A}_{i+1} \circ P2(i) \preceq_{O(n^{-2})} \mathbf{B} \circ P2(i)$ *for all* $i \in \{0,\ldots,m-1\}$*, as long as* B_{i+1} *is* β*-balanced,* $|B_{i+1}|$ *satisfies the bound given in Theorem 2.1, and* $m = \alpha n$*.*

Proof of Theorem 2.1: We will show that $P1(i) \preceq_{\varepsilon_i} P2(i)$ for all i, where $\varepsilon_i = O\left(\frac{i}{n^2}\right)$. This implies the theorem as follows: since $mk < n$, $P2(m)$ has maximum load 1; since $P2(m)$ ε_m-dominates $P1(m)$, $P1(m)$ must have the same maximum load with probability $\geq 1 - \varepsilon_m = 1 - O(m/n^2) = 1 - O(n^{-1})$.

We show $P1(i) \preceq P2(i)$ by induction on i. For the base case where $i = 0$, note that $P1(0) = P2(0)$ trivially, since no balls have been placed. For the inductive step, assume that $P1(i) \preceq_{\varepsilon_i} P2(i)$. Then we have

$$\begin{array}{rcll} P1(i+1) & = & \mathbf{A}_{i+1} \circ P1(i) & \text{(by definition)} \\ & \preceq_{\varepsilon_i} & \mathbf{A}_{i+1} \circ P2(i) & \text{(induction and Lemma 2.1)} \\ & \preceq_{O(n^{-2})} & \mathbf{B} \circ P2(i) & \text{(Lemma 2.2)} \\ & = & P2(i+1) & \text{(by definition).} \end{array}$$

Letting $\varepsilon_{i+1} = \varepsilon_i + O(n^{-2})$, we can conclude that $P1(i+1) \preceq_{\varepsilon_{i+1}} P2(i+1)$.

For the case $m = n$, let D_t be the allocation where each bin has exactly t balls. By the above result, placing αn balls results in an allocation $O(n^{-2})$-dominated by D_1. It is easy to see that beginning with D_1 and placing αn additional balls results in an allocation $O(n^{-2})$-dominated by D_2; iterating this argument yields the desired result for $m = n$. ∎

2.2 Proof of Lemma 2.1. Algorithm $\mathbf{A}_i$ is equivalent to (1) picking a random set of bins B_i, (2) picking a uniform-random permutation $(b_1,\ldots,b_\ell)$ of the bins B_i, and (3) deterministically placing the ball in the *first least-loaded bin*, i.e., the first bin in the permutation among those with the minimum number of balls. Now let (f, f') be a coupling which witnesses $D \preceq_\varepsilon E$. To show that $\mathbf{A}_i \circ D \preceq_\varepsilon \mathbf{A}_i \circ E$, we construct a coupling (g, g') of $\mathbf{A}_i \circ D$ and $\mathbf{A}_i \circ E$ by coupling the randomness in D and E in the same way as (f, f'), and additionally coupling $\mathbf{A}_i$'s choice of B_i and $(b_1,\ldots,b_\ell)$.

We now show that this coupling witnesses $\mathbf{A}_i \circ D \preceq_\varepsilon \mathbf{A}_I \circ E$. Consider any outcome in which $D_j \leq E_j$ for all bins j. Let j^* be the bin in which the ball is placed in $\mathbf{A}_i \circ D$, that is, the first least-loaded bin in $(b_1,\ldots,b_\ell)$. Since $D_j \leq E_j \forall j$, the first least-loaded bin for $\mathbf{A}_i \circ E$ must either occur at j^* or at some later bin in the permutation—so in $\mathbf{A}_i \circ E$ the ball is either placed in the same bin j^* or else j^* already had more balls than in $\mathbf{A}_i \circ D$. In either case, $(\mathbf{A}_i \circ D)_{j^*} \leq (\mathbf{A}_i \circ E)_{j^*}$ which further implies $(\mathbf{A}_i \circ D)_j \leq (\mathbf{A}_i \circ E)_j \forall j$. Finally, the assumption that $D_j \leq E_j \forall j$ is true for a set of outcomes of measure $\geq 1 - \varepsilon$, from which we can conclude $\mathbf{A}_i \circ D \preceq_\varepsilon \mathbf{A}_i \circ E$.

2.3 Proof of Lemma 2.2. For notational convenience, we use $\mathbf{A} := \mathbf{A}_{i+1}$ throughout this section.

We prove Lemma 2.2 using several supporting lemmas. First, we provide an analog of Hall's Theorem for fractional matchings with vertex weights (Lemma 2.3), which we use to show that $\mathbf{A} \circ D \preceq \mathbf{B} \circ D$ for any fixed allocation D that obeys a "smoothness" property (Lemma 2.4). Next, we show that the fraction of bins which are empty in the set B_i chosen by algorithm $\mathbf{A}$ concentrates (Lemma 2.5), which implies that $P2(i)$ is smooth with high probability (Lemma 2.6). Finally, Lemmas 2.4 and 2.6 together imply Lemma 2.2, which we prove at the end of this subsection.

DEFINITION 2.4. *Given an undirected bipartite graph* $G = (V_1, V_2, E)$ *and weights* $w(v) \geq 0$ *for each vertex* v*, a* perfect weighted matching *on* G *is a nonnegative function* $f : E \to \mathbb{R}$ *for which*

$$\forall v \in V_1, V_2 \quad f(v) = w(v), \tag{2.1}$$

where $f(v) := \sum_{e \sim v} f(e)$*.*

LEMMA 2.3. *Given a bipartite graph* $G = (V_1, V_2, E)$ *with vertex weights* w *for which* $w(V_1) = w(V_2)$*, a perfect weighted matching exists if and only if* $w(S) \leq w(\mathcal{N}(S))$ *for all* $S \subseteq V_1$ *(equivalently, for all* $S \subseteq V_2$*), where* $w(S) := \sum_{v \in S} w(v)$ *and* $\mathcal{N}(v)$ *is the neighborhood of* v*.*

Proof: Follows a standard max flow-min cut argument; see Appendix A. ■

DEFINITION 2.5. *A fixed allocation D of balls to bins is* β-smooth *when, for each bin j,*

$$\frac{1}{\beta fn} \cdot 1_{(D_j=0)} \leq p_j \leq \frac{\beta}{fn} \cdot 1_{(D_j=0)} + O(n^{-3}) \cdot 1_{(D_j>0)},$$

where p_j is the probability that Algorithm **A** *places a ball in bin j given the allocation D, f is the fraction of bins that are empty in D, and D_j denotes the number of balls in bin j.*

The smoothness property says essentially that **A** picks one of the fn empty bins with probability that is within a factor β of the "fair chance", i.e., $\frac{1}{fn}$, and has only a very small chance of picking an occupied bin. Thus, smoothness depends on both the allocation D and the placement algorithm **A**.

LEMMA 2.4. *Let D be a β-smooth allocation of ki balls. Then $\mathbf{A} \circ D \preceq_{O(n^2)} \mathbf{B} \circ D$ as long as $k \geq \left\lceil 1 - \frac{\ln \beta}{\ln(1-(\beta-1)/(\beta^2-1))} \right\rceil$.*

Proof: To demonstrate $\mathbf{A} \circ D \preceq_\varepsilon \mathbf{B} \circ D$, it is sufficient to show that there is a measure-preserving mapping f between outcomes in $\mathbf{A} \circ D$ and outcomes in $\mathbf{B} \circ D$ such that $\omega_j \leq f(\omega)_j$ for all outcomes ω, except on a set of measure $\leq \varepsilon$ (where ω_j represents the number of balls in bin j in outcome ω). To show that such a mapping exists, we will construct a suitable bipartite graph $G = (V_1, V_2, E)$ and weights $w(\cdot)$, and find a perfect weighted matching within it, as follows. We have a node in V_1 for each subset of k bins drawn from the empty bins in D. The weight $w(v)$ of a node $v \in V_1$ is the probability that the set v of bins is selected to receive balls in $\mathbf{B} \circ D$. Each node in $j \in V_2$ corresponds to a single empty bin in D, and its weight $w(j)$ is set to p_j, the probability that bin j receives the ball in $\mathbf{A} \circ D$. We add an additional node j^* to V_2 with weight $w(j^*) = \Pr[\mathbf{A}$ puts a ball in an occupied bin$]$. The graph contains an edge $v \to j$ whenever $j \in v$, and an edge $v \to j^*$ for all v.

Note that if $(v, j) \in E$ and $j \neq j^*$, then outcome v has a ball in every bin in which j has a ball. As a consequence, the existence of a perfect weighted matching on G with weights w implies that $\mathbf{A} \circ D \preceq_{w(j^*)} \mathbf{B} \circ D$. Moreover by a union bound over the n bins and the fact that D is smooth, $w(j^*) = O(n \cdot n^{-3}) = O(n^{-2})$, as desired.

It remains to show that a perfect weighted matching exists, which we accomplish by checking the sufficient condition, $w(S) \leq w(\mathcal{N}(S)) \, \forall S \subseteq V_2$, given in Lemma 2.3. Consider any subset $S \subseteq V_2$ of the nodes. If $j^* \in S$, then the condition is trivially satisfied since $\mathcal{N}(S) = V_1$ and $w(V_1) = w(V_2)$ by construction. Thus, we may assume $j^* \notin S$. For convenience define $f = \frac{n-ki}{n}$ to be the fraction of bins that are empty. To upper bound $w(S)$, we have

$$w(S) \leq \frac{\beta \cdot |S|}{fn} \tag{2.2}$$

$$w(S) = 1 - w(V_2 \setminus S) \leq 1 - \frac{fn - |S|}{\beta fn}, \tag{2.3}$$

both of which follow from the β-smoothness of D. Also,

$$\begin{aligned} w(\mathcal{N}(S)) &= \Pr[\text{one of } k \text{ random bins hits } S] \\ &= 1 - \left(\frac{fn - |S|}{fn}\right)\left(\frac{fn - |S| - 1}{fn}\right) \cdots \\ &\quad \cdots \left(\frac{fn - |S| - (k-1)}{fn}\right). \end{aligned}$$

Note that $w(\mathcal{N}(S))$ is concave in $|S|$ and reaches 1 for sufficiently large $|S|$, and both Eqns. 2.2 and 2.3 are linear in $|S|$. Letting s^* be the value of $\frac{|S|}{fn}$ for which the upper bounds (2.2) and (2.3) are equal, it thus suffices to show that $w(S) \leq w(\mathcal{N}(S))$ when $|S|/n = s^*$. In this case we have

$$\begin{aligned} w(\mathcal{N}(S)) &\geq 1 - \left(\frac{fn - |S|}{fn}\right)^k = 1 - (1 - s^*)^k \\ w(S) &\leq 1 - \frac{fn - |S|}{\beta fn} = 1 - \frac{1 - s^*}{\beta}. \end{aligned}$$

Solving $1 - \frac{1-s^*}{\beta} \leq 1 - (1 - s^*)^k$, we obtain $k \geq 1 - \frac{\ln \beta}{\ln(1-s^*)}$. Substituting $s^* = \frac{\beta-1}{\beta^2-1}$ yields the lemma. ■

The above lemma showed essentially that smoothness is sufficient for Lemma 2.2. It remains to show that $P2(i)$ is smooth. Smoothness requires that the chance that **A** places a ball in a particular bin is nearly equal for all empty bins, but this probability is inversely proportional to the number of empty bins in B_{i+1}. Our strategy is therefore to show that with high probability, the number of empty bins in B_{i+1} is close to its mean; that is, the probed set is "good".

We proceed by defining "goodness", showing that any fixed set of bins is very likely good (Lemma 2.5), and then extending this to show that $P2(i)$ is smooth (Lemma 2.6). At the end of this section we bring together the building blocks into a proof of Lemma 2.2.

DEFINITION 2.6. *A set of bins B is* ε-good *if the number of empty bins in B is contained in $\left[\frac{1}{1+\varepsilon} \cdot f|B|, (1+\varepsilon) \cdot f|B|\right]$, where $f = \frac{n-ki}{n}$ is the (random) fraction of bins that are empty in $P2(i)$.*

LEMMA 2.5. *For any fixed set of bins B for which $|B| \geq \frac{(4\ell+2)(1+\varepsilon)^2}{f\varepsilon^2} \cdot \ln n$ and any $\ell, \varepsilon > 0$, $\Pr_{P2(i)}[B \text{ is } \varepsilon\text{-good}] \geq 1 - O(n^{-\ell})$.*

Proof: We use a "Poissonization" technique to convert from $P2$ to a distribution wherein the bins are independent (see [9]). Define a new distribution $P3(i)$ which places a ball in each bin independently with probability ik/n, and let F be the (random) fraction of bins that are empty in $P3(i)$. Note that $P3(i)$ may have more or fewer balls than $P2(i)$, but conditioned on $F = f$, $P3(i)$ is identical to $P2(i)$. Moreover, $\mathrm{E}[F] = f$ and Fn is approximately Poisson, so $\Pr_{P3(i)}[F = f] = \Omega\left(\frac{1}{\sqrt{n}}\right)$. Using these facts, and defining for convenience $\delta = \frac{\varepsilon}{1+\varepsilon}$ and $G = \{B \text{ is } \varepsilon\text{-good}\}$,

$$\begin{aligned}
\Pr_{P2(i)}[\neg G] &= \Pr_{P3(i)}[\neg G \mid F = f] \\
&\leq \frac{\Pr_{P3(i)}[\neg G]}{\Pr_{P3(i)}[F = f]} \\
&\leq \frac{\Pr_{P3(i)}\left[F \notin \left[\frac{1}{1+\varepsilon} f|B|,\ (1+\varepsilon) f|B|\right]\right]}{\Omega(1/\sqrt{n})} \\
&\leq O\left(\sqrt{n}\right) \cdot \Pr_{P3(i)}[F \notin (1 \pm \delta) f|B|] \\
&\leq O\left(\sqrt{n}\right) \cdot \left(e^{-f|B|\delta^2/4} + e^{-f|B|\delta^2/2}\right),
\end{aligned}$$

by applying a pair of Chernoff bounds [10]. This bound is $\leq O(n^{-\ell})$ as long as $|B| \geq \frac{(4\ell+2)(1+\varepsilon)^2}{f\varepsilon^2} \cdot \ln n$. ∎

LEMMA 2.6. *$\Pr[P2(i)$ is $(1 + \varepsilon + o(1))\beta$-smooth$] \geq 1 - O(n^{-2})$ for any $\varepsilon > 0$ as long as $|B_{i+1}| \geq \frac{26(1+\varepsilon)^2}{f\varepsilon^2} \cdot \ln n$ and $\mathbf{A}$ is β-balanced.*

Proof: Recall from Definition 2.5 that to show smoothness of $P2(i)$, we must bound the probability p_j that $\mathbf{A}$ puts a ball in bin j. Here p_j is itself a random variable, dependent on $P2(i)$. For convenience let $B := B_{i+1}$ be the random set of bins selected by $\mathbf{A}$ for ball $i+1$, let G be the event that B is ε-good, and let P_j be the event that j receives the ball. For the upper bound, we have

$$\begin{aligned}
p_j &= \Pr[j \in B] \cdot \Pr[P_j | j \in B] \\
&= \Pr[j \in B] \cdot (\Pr[P_j \wedge G | j \in B] + \\
&\quad \Pr[P_j \wedge \neg G | j \in B]) \\
&\leq \Pr[j \in B] \cdot \Pr[\neg G | j \in B] + \Pr[j \in B] \sum_{s=1}^{n} \\
&\quad \Pr[P_j \wedge G | j \in B \wedge |B| = s] \cdot \Pr[|B| = s | j \in B] \\
&\leq \Pr[\neg G | j \in B] + \Pr[j \in B] \cdot 1_{(j \text{ empty})} \sum_{s=1}^{n} \\
&\quad \frac{1+\varepsilon}{fs} \cdot \Pr[|B| = s | j \in B] \quad \text{(by def. of } \varepsilon\text{-good)} \\
&= \Pr[\neg G | j \in B] + \frac{1+\varepsilon}{f} \cdot 1_{(j \text{ empty})} \cdot \\
&\quad \Pr[j \in B] \cdot \mathrm{E}\left[\frac{1}{|B|} | j \in B\right] \\
&\leq \Pr[\neg G | j \in B] + \frac{(1+\varepsilon)\beta}{fn} \cdot 1_{(j \text{ empty})},
\end{aligned}$$

where the last step follows since $\mathbf{A}$ is β-balanced. By Lemma 2.5 and the fact that $\mathbf{A}$ picks B independently of $P2(i)$, for any bin j, $\Pr_{\mathbf{A} \circ P2(i)}[\neg G | j \in B] = O(n^{-6})$ as long as $|B| \geq \frac{26(1+\varepsilon)^2}{f\varepsilon^2} \cdot \ln n$. Thus,

$$\Pr_{P2(i)}\left[\Pr_{\mathbf{A}}[\neg G | j \in B] \leq O(n^{-3})\right] \geq 1 - O(n^{-3}).$$

Combining the previous two inequalities, we obtain an upper bound on p_j:

$$\Pr_{P2(i)}\left[p_j \leq \frac{(1+\varepsilon)(1+o(1))\beta}{fn} \cdot 1_{(j \text{ empty})} + O(n^{-3}) \cdot 1_{(j \text{ occupied})}\right] \geq 1 - O(n^{-3}).$$

Using a similar technique we can obtain the lower bound,

$$\Pr_{P2(i)}\left[p_j \geq \frac{1_{(j \text{ empty})}}{(1+\varepsilon)(1+o(1))\beta fn}\right] \geq 1 - O(n^{-3}).$$

Combining the upper and lower bounds and taking a union bound over all n bins j, we have that $P2(i)$ is $(1 + \varepsilon + o(1))\beta$-smooth with probability $\geq 1 - O(n^{-2})$. ∎

Proof of Lemma 2.2: Let $\beta' := (1 + \varepsilon + o(1)) \cdot \beta$. Construct the coupling of $\mathbf{A} \circ P2(i)$ and $\mathbf{B} \circ P2(i)$ as follows. For each outcome (allocation) for which $P2(i)$ is β'-smooth, use the coupling given by Lemma 2.4. Conditioned on such an outcome D, we have $\mathbf{A} \circ D \preceq_{O(n^{-2})} \mathbf{B} \circ D$. By Lemma 2.6, this covers all but a set of measure $O(n^{-2})$, on which set algorithms $\mathbf{A}$ and $\mathbf{B}$ may be coupled arbitrarily, yielding $\mathbf{A} \circ P2(i) \preceq_{O(n^{-2})+O(n^{-2})} \mathbf{B} \circ P2(i)$. To satisfy the constraints of these two lemmas, we require $k \geq \left\lceil 1 - \frac{\ln \beta'}{\ln(1-(\beta'-1)/(\beta'^2-1))} \right\rceil$ and $|B| \geq \frac{26(1+\varepsilon)^2}{f\varepsilon^2} \cdot \ln n$. The fraction f of empty bins in $P2$ must in turn be $\geq \delta$ for some $\delta > 0$, which is satisfied as long as $\alpha \leq \frac{1-\delta}{k}$, as given in the statement of the lemma. ∎

3 Lower Bound

THEOREM 3.1. *For any $d \in \left[1, \frac{\ln n}{\ln \ln n}\right]$, there exists a 1-balanced distribution of sets of bins (B_i) for which $|B_i| = d$ and which results in a maximum load of $\geq \frac{1}{d} \cdot \frac{\ln n}{\ln \ln n} \cdot (1 + o(1))$ w.h.p. when n balls are allocated to the n bins.*

Proof: We will in fact give a class of bin choices which result in the desired maximum load. Given an arbitrary fixed pattern of d bin indexes $B = (b_1, \ldots, b_d)$, we have ball i pick a single random value $R \in \{1, n\}$, and set $B_i = \{b_1 + R, \ldots, b_d + R\}$, with arithmetic mod n. It is easy to see that this distribution is 1-balanced. Now by reduction to a standard balls-and-bins process with $d = 1$ choices, some value r^* of R will be picked by $\frac{\ln n}{\ln \ln n} \cdot (1 + o(1))$ balls w.h.p. [3]. Each of those balls is therefore choosing from the same set of bins $\{b_1 + r^*, \ldots, b_d + r^*\}$, so at least one of those bins receives $\geq \frac{1}{d} \cdot \frac{\ln n}{\ln \ln n} \cdot (1 + o(1))$ balls. ∎

4 Applications

4.1 Nearby Server Selection. We give an example application of our result to a *nearby server selection* problem. We have n servers (e.g., wireless base stations) placed randomly in the unit square in the plane, and m clients (e.g., wireless-enabled laptops) arrive iteratively at random locations, probe the load of a set of servers within some distance r, and connect to a random one of the least loaded among these. We desire to minimize the maximum load on a server, while keeping r small. This problem was suggested to us by [7], as a variant of a client-server matching problem in [4] in which clients may be moved to different servers after they have connected, and the number of moves is minimized. Here we show that in this randomized version of the problem, $\Theta(n)$ clients can be matched to n servers with no moves.

COROLLARY 4.1. *If $r = \Theta\left(\sqrt{\frac{\log n}{n}}\right)$, then with probability $1 - O(n^{-1})$, each client is matched with a unique server as long as $m \leq \alpha n$ for some constant α; and each server has $\leq \lceil 1/\alpha \rceil$ clients if $m = n$.*

Proof: Divide the unit square into subsquares of size $b\sqrt{\frac{\log n}{n}}$ on each side. By a Chernoff bound, there exists a sufficiently large b such that all subsquares contain $\Theta(\log n)$ servers w.h.p. By selecting $r = c\sqrt{\frac{\log n}{n}}$ for sufficiently large c, any possible disc of radius r centered in the unit square will include $\Theta(c^2)$ subsquares and hence $\Theta(c^2 \log n)$ servers in its set of options B_i. Thus, for sufficiently large c, each B_i is large enough to satisfy the lower bound required by Theorem 2.1. Moreover, for any server j, accounting for edge effects,

$$\frac{1}{4} \cdot \frac{c^2 \log n}{n} \leq \Pr[j \in B_i] \leq \frac{c^2 \log n}{n},$$

and by the above argument, w.h.p. over the choice of server locations, $E\left[\frac{1}{|B_i|} | j \in B_i\right] = \Theta\left(\frac{1}{c^2 \log n}\right)$. From this it follows that B_i is $\Theta(1)$-balanced w.h.p. Thus, we can apply Theorem 2.1 and the result follows. ∎

4.2 Load Balance in Distributed Hash Tables. In this section we describe an application of our main result to load balance in distributed hash tables (DHTs). Compared with a similar scheme of Byers et al [2], we will obtain a better balance by using more choices for each file's location, while using the same number of messages (within constant factors) because those choices align with the DHT's structure. We note in advance, however, that unlike [2], our scheme requires a deterministic overlay topology. In particular we will describe our scheme in the context of the Chord DHT [11].

In Chord, each node v is assigned a pseudorandom *identifier* $id(v)$ in the DHT's *keyspace* $[0, 1]$. Ownership of the keyspace is partitioned among the nodes such that a key k is owned by its *successor*—that is, the node whose ID most closely follows k, where the keyspace is treated modularly. Each node maintains links to certain other nodes as a function of the IDs of the nodes. Specifically, each node v has links, called the *successor list*, to the $\Theta(\log n)$ successors of $id(v)$, where n is the number of nodes in the system. Node v also has links called *fingers* to the owners of $id(v) + b^{-i}$ for each $i > 1$, which results in $\Theta(\log_b n)$ additional links. (In the original Chord design, $b = 2$.) Finally, each file o is stored in the DHT at the node that owns the key $h(o) \in [0, 1]$, where h is a well-known hash function. We will assume that n objects are placed.

One challenge is to achieve a good load balance. Due to the random identifier selection, some nodes own $\Theta(\frac{\log n}{n})$ of the keyspace and hence can expect to receive $\Theta(\log n)$ objects. Even if all nodes had equal shares of the keyspace, the standard balls-and-bins result would imply a maximum load of $\Theta(\frac{\log n}{\log \log n})$.

Byers et al [2] showed that if we have d well-known hash functions $h_1, \ldots, h_d$ and each ball is placed at the least-loaded among the owners of $h_1(o), \ldots, h_d(o)$, then the maximum load is $\frac{\log \log n}{\log d} + O(1)$ w.h.p.—even though some nodes (bins) are a factor $\Theta(\log n)$ more likely to be considered than others. Increasing d obtains a better load balance, but at the cost of requiring more messages to insert and retrieve objects. Specifically, since mean route lengths in Chord are $\Theta(\log n)$ [11], $\Theta(d \log n)$ messages will be sent on average.

Our scheme will have a single hash function h, and each object o will be placed in a random least-loaded node among a set of nodes $B(o)$ chosen as the owners of $h(o) + b^{-i}$, for each $i > 1$. This maintains the essential property that each file is located at one of a small number of well-known locations, but mirrors the topology of the DHT. We next analyze the messaging cost and maximum load of this scheme. In the interest of conciseness, we only sketch the analysis.

The scheme can be implemented by routing an

insert request to the owner x of $h(o)$ at an expected cost of $\Theta(\log n)$ messages. The node x then forwards the request to the nodes $B(o)$. Since $|h(o) - x| = O\left(\frac{\log n}{n}\right)$ w.h.p., node x has a direct connection (via a finger link) to a node within keyspace distance $O\left(\frac{\log n}{n}\right)$ of each probe target. By using the DHT's successor list, those neighbors can reach the probe targets in $O(1)$ additional hops each. Thus, $O(\log n)$ messages are sent in total.

In bounding the maximum load, we apply Theorem 2.1. To deal with the imbalance in the size of the keyspace that nodes own, we will simply have each object ignore any node in B whose ownership is $< \frac{1}{cn}$ or $> \frac{c}{n}$ for some constant c. Note that this new strategy can only increase the maximum load. It can be shown that this will eliminate a constant fraction of the object's choices, leaving $\Theta(\log n)$ choices w.h.p.; moreover, the constant can be made arbitrarily high by adjusting b so that Theorem 2.1's constraint on $|B|$ is satisfied. It is also straightforward to show that for any node v owning a fraction $\in [\frac{1}{cn}, \frac{c}{n}]$ of the keyspace, $\Pr[v \in B] = \Theta\left(\frac{\log n}{n}\right)$. Thus, we can apply Theorem 2.1 to conclude that the maximum load is $O(1)$ w.h.p.

5 Conclusion

This paper leaves several open problems. A factor $\log \log n$ gap remains between our lower and upper bounds. Also, we conjecture that for any $\varepsilon > 0$, it is possible to place $(1 - \varepsilon)n$ balls while maintaining a maximum load of 1 w.h.p., as long as $d = \Theta(\log n)$ is sufficiently large and the probed bins B_i are sufficiently close to being 1-balanced.

Our model requires that each ball be placed in a *uniform-random* least-loaded bin among those probed. Suppose instead that each ball arrives with an ordered list of the bins, and is placed in the first empty bin on the list (assuming there are $m \leq n$ balls). What properties of the orderings ensure that only the first few bins on each list need to be probed? Note that linear probing in hash tables [6] is a special case of this model in which the ordered lists are given by $\{R, R+1, \ldots, R+n-1\} \pmod n$ where R is uniform-random in $\{1, \ldots, n\}$.

We thank the anonymous reviewers for useful comments and corrections.

References

[1] Y. Azar, A. Z. Broder, A. R. Karlin, and E. Upfal. Balanced allocations. In *Proc. STOC*, 1994.

[2] John Byers, Jeffrey Considine, and Michael Mitzenmacher. Geometric generalizations of the power of two choices. In *Proc. SPAA*, 2004.

[3] Gaston Gonnet. Expected length of the longest probe sequence in hash code searching. In *Journal of the ACM*, volume 28, April 1981.

[4] Edward F. Grove, Ming-Yang Kao, P. Krishnan, and Jeffrey Scott Vitter. Online perfect matching and mobile computing. In *Proceedings of the Fourth Workshop on Algorithms and Data Structures (WADS)*, 1995.

[5] Krishnaram Kenthapadi and Rina Panigrahy. Balanced allocation on graphs. In *Proc. SODA*, 2006.

[6] Donald E. Knuth. *Art of Computer Programming, Volume 3: Sorting and Searching (2nd Edition)*. Addison-Wesley Professional, April 1998.

[7] Henry Lin, Constantinos Daskalakis, Robert Kleinberg, and Kamalika Chaudhuri. Personal communication, 2007.

[8] Michael Mitzenmacher. *The Power of Two Choices in Randomized Load Balancing*. PhD thesis, University of California - Berkeley, 1996.

[9] Michael Mitzenmacher and Eli Upfal. *Probability and Computing*. Cambridge University Press, 2005.

[10] Rajeev Motwani and Prabhakar Raghavan. *Randomized Algorithms*. Cambridge University Press, 1995.

[11] Ion Stoica, Robert Morris, David Karger, M. Frans Kaashoek, and Hari Balakrishnan. Chord: A scalable peer-to-peer lookup service for internet applications. In *Proc. SIGCOMM*, 2001.

[12] Berthold Vöcking. How asymmetry helps load balancing. In *IEEE Symposium on Foundations of Computer Science*, pages 131–141, 1999.

[13] Udi Wieder. Balanced allocations with heterogenous bins. In *Proc. SPAA*, 2007.

A Proof of Lemma 2.3

The necessity of the condition is clear. We next show that a perfect weighted matching exists assuming the condition holds. Construct a graph H which includes the nodes and edges of $G = (V_1, V_2, E)$, additional source and sink nodes s and t, an edge $s \to v$ with capacity $w(v)$ for each $v \in V_1$, and an edge $v \to t$ with capacity $w(v)$ for each $v \in V_2$. Finally, we give each edge in E capacity ∞.

It is easy to see that there is a perfect matching in G if and only if the maximum flow $f_{\max}$ from s to t in H is $w(V_1)$. Thus for the remainder of the proof it is sufficient to show that if $f_{\max} < w(V_1)$ then there exists an $S \subseteq V_1$ for which $w(\mathcal{N}(S)) < w(S)$.

Suppose $f_{\max} < w(V_1)$, and let (C_1, C_2) be a minimum cut of H with $s \in C_1$ and $t \in C_2$. We take $S := C_1 \cap V_1$. Since the edges of E have capacity ∞, C_1 must include $\mathcal{N}(S)$. Therefore the cut edges must be those from s to $V_1 \setminus S$, and those from $\mathcal{N}(S)$ to t. The total capacity of these edges is $w(V_1) - w(S) + w(\mathcal{N}(S))$. Thus we have $w(V_1) - w(S) + w(\mathcal{N}(S)) = f_{\max} < w(V_1)$, so $w(\mathcal{N}(S)) < w(S)$ as desired.