Multidimensional Balanced Allocations

Andrei Broder* Michael Mitzenmacher†

Abstract

We consider a multidimensional variant of the balls-and-bins problem, where balls correspond to random D-dimensional 0-1 vectors. This variant is motivated by a problem in load balancing documents for distributed search engines. We demonstrate the utility of the power of two choices in this domain.

1 Introduction

When n balls are sequentially placed uniformly at random into n bins, the fullest bin has $(1+o(1)) \ln n / \ln \ln n$ balls in it w.h.p. (with high probability). In contrast, Azar *et. al.* [1] have shown that, if for each ball we choose d bins at random and then place the ball into the least full of them at the time of insertion, then w.h.p. the fullest bin contains only $\ln \ln n / \ln d + O(1)$ balls. This idea, referred to as "balanced allocations" or "the power of two choices," has spawned a large literature of both theoretical and practical interest [4].

Here we consider a multidimensional variation: Consider throwing m balls into n bins, where each ball is a random D-dimensional 0-1 vector of weight f; that is, each vector has exactly f non-zero entries chosen uniformly among all $\binom{D}{f}$ possibilities. The average load in each dimension for each bin is then $\alpha = (mf)/(nD)$. Let $\ell(a,b)$ be the load in dimension a for the b-th bin. The maximum *dimensional load* (as opposed to the maximum load over bins) is $\max_{a,b} \ell(a,b)$. We show that under natural conditions placing the balls randomly or in a round-robin fashion yields a maximum dimensional load of $\Omega(\log(nD)/\log\log(nD))$, while using the power of two choices appropriately can reduce the maximum dimensional load to $O(\log\log(nD))$.

The motivation for our problem arises from large-scale search engines. Because the collection of pages to be indexed is so large, it has to be split among n servers. When a user makes a query to a front-end machine, the query is sent to all n servers; results are returned to the front-end machine for merging and presentation. Hence the time to serve the query is determined by the slowest of the servers. The time for each server to process a one-word query is roughly proportional to the number of documents at that server containing the word of interest. Thus, to achieve better efficiency, it is necessary to split the documents among servers in such way that the number of documents containing a given word is roughly equal.

We are of course simplifying or ignoring many issues, including: (a) the time to serve a one-word query is usually proportional to the number of disk reads needed to read its *posting list*, i.e. the list of its occurrences in the corpus; (b) for multiple word queries, one can use branch-and-bound techniques (see e.g. [2]); (c) it is possible to simply ignore the slowest responding servers, with a corresponding decline in answer quality (see [3]). Nevertheless, at least for popular words, it is a good idea to split the documents among servers in such a way as to balance these words as much as possible.

In our setting, the n bins represent the n servers and m represents the total number of documents in the collection. The dimension D represents the number of "interesting words" that should be balanced. Here D is much smaller than the full vocabulary: most words have posting lists that can be read in one disk read and require no optimization, and words that appear very seldom in queries are of no interest as well. The weight f is the maximum number of distinct interesting words that might appear in any document.

We present work in progress, providing some theoretical analysis and simulation results. We believe the study of the power of two choices in multidimensional settings is interesting in its own rights, and will find further applications.

2 A Lower Bound

We first provide a lower bound that holds when the m balls are placed at random, or when the balls are simply split among the n bins in a round-robin fashion.

LEMMA 2.1. *Conditional on the load in each bin being m/n and $f/D < 1/2$,*

$$\Pr(\ell(a,b) \geq k) \geq (2^{-2\alpha})(\alpha/k)^k.$$

Proof.

$$\begin{aligned}
\Pr(\ell(a,b) \geq k) &= \sum_{j=k}^{m/n} \binom{m/n}{j} \left(\frac{f}{D}\right)^j \left(1 - \frac{f}{D}\right)^{m/n-j} \\
&\geq \binom{m/n}{k} \left(\frac{f}{D}\right)^k \left(1 - \frac{f}{D}\right)^{m/n} \\
&\geq (m/nk)^k (f/D)^k (1 - f/D)^{\alpha D/f} \\
&\geq (2^{-2\alpha})(\alpha/k)^k.
\end{aligned}$$

*IBM T. J. Watson Research Center, 19 Skyline Dr., Hawthorne, NY 10532. E-mail: abroder@us.ibm.com.

†Harvard University, Computer Science Department, 33 Oxford St., Cambridge, MA 02138. Supported by NSF grants CCR-9983832, CCR-0118701, and CCR-0121154. E-mail: michaelm@eecs.harvard.edu.

LEMMA 2.2. *When α is a constant and $m/n > \log(nD)/\log\log(nD)$, the maximum dimensional load is $\Omega(\log(nD)/\log\log(nD))$ with high probability.*

Proof. We sketch the proof. The probability that the maximum dimensional load exceeds any given value x is minimized when all bins have m/n balls. In this case, the events $\ell(a,b) \geq x$ are easily shown to be negatively dependent, so that

$$\Pr(\forall a, b : \ell(a,b) < x) \leq (1 - (2^{-2\alpha})(\alpha/x)^x)^{nD},$$

using Lemma 2.1. The right hand side can be shown to be less than $1/n$ for an $x = \Omega(\log(nD)/\log\log(nD))$.

3 An Upper Bound

Our upper bound using the power of two choices is limited to cases where f is polylogarithmic in n. We expect to have more general results in an extended version of this paper. In our scheme, d bins $b_1, b_2, \ldots, b_d$ are chosen uniformly at random. Let S be the set of values set to 1 in the vector being placed. Then the values $\max_{a \in S} \ell(a, b_i)$ are compared. The ball is put in the bin with the smallest corresponding value, ties being broken arbitrarily.

THEOREM 3.1. *When α and d are constants, f is $O(\log^c n)$, and D is polynomial in n, then the maximum dimensional load is $\log\log(nD)/\log d + o(\log\log(nD))$ with high probability.*

Proof. We sketch the proof, following the layered induction technique introduced in [1]. Let β_i represent an upper bound that holds w.h.p. on the number of dimension-bin pairs (a,b) with load at least i. We want a recurrence for the β_i that holds inductively w.h.p. We set

$$\beta_{j+1} = 2fm\left(\frac{f\beta_j}{nD}\right)^d.$$

To see the reason for this choice, notice that if the i-th bin is chosen,

$$\Pr(\max_{a \in S} \ell(a, b_i) \geq j) \leq (f\beta_j)/(nD),$$

and hence the probability that all d choices match a dimension with load at least j is at most $((f\beta_j)/(nD))^d$. When this occurs, it introduces at most f dimensions with load at least $j+1$. Hence the expected number of dimensions with load at least $j+1$ is at most $fm((f\beta_j)/(nD))^d$, and we add a factor of two so that the appropriate Chernoff bound holds with high probability. We conclude that we may set

$$\frac{\beta_{j+1}}{nD} = 2\alpha\left(\frac{f\beta_j}{nD}\right)^d,$$

so that the fraction of dimensions with load at least j shrinks rapidly.

With this recurrence, it remains to show that:

1. There is a suitable starting point j for the recurrence;
2. The recurrence has β_j shrink to a small number when $j = \log\log n/\log d + o(\log\log(nD))$;
3. The tail of the argument (once β_j is so small that the Chernoff bound does not apply) causes no difficulties.

The tail is easily handled using standard means. The start of the recurrence is more challenging; due to the extra factors of f, we must be careful to ensure that β_j is initially decreasing. However, if we let $j_0 = c_1 \log\log n/\log\log\log n$ for a suitable constant c_1, we can show that $\beta_{j_0} \leq nD/\log^{2c} n$. To see this, suppose that every ball is placed d times, according to all of its d choices. Even in this case, the number of dimensions with load at least j is essentially the same as when $dmf = d\alpha nD$ balls are thrown into nD bins, implying $\beta_{j_0} \leq nD/\log^{2c} n$.

Given the initial value β_{j_0}, standard techniques show that β_j falls fast enough that for $k = \log\log(nD)/\log d + j_0$, $\beta_k \leq \log n$, at which point we have to use a tail argument to show that there is no dimension with load more than $k = \log\log(nD)/\log d + j_0 + O(1)$ with high probability.

4 Simulation Example

As an example, we performed simulations with 10,000 bins and 10,000,000 balls, each ball corresponding to a 1,000 dimensional 0-1 vector with 20 random entries set to 1, so that $\alpha = 20$. We evaluated the power of two choices compared with a single random choice and a round-robin division. Over 100 trials, the maximum dimensional load using two choices was always 29 or 30; using one choice it ranged between 46 and 52; and using round-robin it ranged between 41 and 49. This is in line with our theory, and suggests the type of improvement one might expect with this simple approach.

References

[1] Azar, Y., Broder, A., Karlin, A., and Upfal, E. Balanced allocations. *SIAM Journal of Computing 29*, 1 (2000), 180–200.

[2] Broder, A., Carmel, D., Herscovici, M., Soffer, A., and Zien, J., Efficient query evaluation using a two-level retrieval process. In *Proc. of the* 12^{th} *CIKM* (2003), pp. 426-434.

[3] Broder, A. and Mitzenmacher, M. Optimal plans for aggregation. In *Proc. of the* 21^{st} *PODC* (2002), pp. 144–152.

[4] Mitzenmacher, M., Richa, A., and Sitaraman, R. The Power of Two Choices: A Survey of Techniques and Results. In *Handbook of Randomized Computing*, edited by P. Pardalos, S. Rajasekaran, J. Reif, and J. Rolim. Kluwer Academic Publishers, Norwell, MA, 2001, pp. 255–312.