Balls and Bins - A Tutorial

Berthold Vöcking
Universität Dortmund
Simple randomized allocation

- Assign $n$ balls to $n$ bins.
- For every ball, choose a bin independently, uniformly at random (iur.).
Let $L$ denote the number of balls in the fullest bin (maximum load).

$$\mathbb{E}[L] = \Gamma^{-1}(n) - \frac{3}{2} + o(1) = (1 + o(1)) \frac{\ln n}{\ln \ln n}.$$  

[Gonnet 1981]

Maximum load is sharply concentrated:

$$L = O \left( \frac{\ln n}{\ln \ln n} \right) \text{ w.h.p.}$$

where w.h.p. abbreviates with prob. at least $1 - \frac{1}{n^\kappa}$, for any fixed $\kappa$. 

Berthold Vöcking, MPI
Motivation

randomized load balancing

- balls = jobs
- bins = machines
- $L = \text{max. number of jobs per machine}$

data allocation

- balls = data items
- bins = memory modules (disks)
- $L = \text{contention}$
**hashing**

- balls = keys
- bins = table positions
- $L = \text{max. chain length}$

**routing**

- balls = requests for connections
- subset of bins = paths in a network
- $L = \text{congestion, i.e., max. number of paths using the same edge}$
Multiple-choice allocation

Idea:

- choose a small sample $S'$ of bins at random
- inspect bins in $S'$ and place ball into one of them
Multiple-choice allocation

Idea:

- choose a small sample $S'$ of bins at random
- inspect bins in $S'$ and place ball into one of them

A small sample of previous work:

- PRAM simulation (Karp, Luby, Meyer auf der Heide, 1992)
- balanced allocation (Azar, Broder, Karlin, Upfal, 1994)
- parallel load balancing (Adler et al., 1995)
- multiple-choice queueing systems (Mitzenmacher, 1996)
- redundant arrays of discs (Sanders, Egner, Korst, 2000)
Balanced Allocation

- balls are inserted one after the other (on-line)
- for every ball, choose $d \geq 2$ alternatives uniformly at random
- give ball to alternative with smallest load
Theorem: Balanced allocations, regardless of tie breaking mechanism, yields

\[ L = \frac{\ln \ln n}{\ln d} \pm \Theta(1) \text{ w.h.p.} \]

Proof:

- \( \alpha_i := \) fraction of bins with at least \( i \) balls after placing the last ball
- \( \alpha_i(b) := \) fraction of bins with at least \( i \) balls at insertion time of ball \( b \)
Let $X_i^b = 1$ if ball $b$ has height $\geq i$, otherwise 0.

Observation: $\Pr [X_i^b = 1] = (\alpha_{i-1}(b))^d \leq (\alpha_{i-1})^d$.

Let $B_i$ denote the number of balls with height $\geq i$.

$E[B_i] = \sum_b \Pr [X_i^b = 1] \leq n (\alpha_{i-1})^d$. 
Proof Idea:

- $E[n \alpha_i] \leq E[B_i] \leq n (\alpha_{i-1})^d$.
- $E[n \alpha_i]$ is sharply concentrated so that we obtain the following "recurrence":

\[
\begin{align*}
\alpha_i &\approx (\alpha_{i-1})^d, \quad \text{for } i \geq 3 \\
\alpha_2 &\leq \frac{1}{2}.
\end{align*}
\]

- Solving the recurrence yields $\alpha_i \approx 2^{-d^i}$.
- For $\ell \approx \log_d \log n$, one obtains $\alpha_\ell \approx \frac{1}{n}$.
- Thus one might guess that the maximum load is $L \approx \log_d \log n$.
- Now let’s go back to more formal arguments ...
Adversary Model:

- Suppose an adversary can mark bins in such a way that at most an $\alpha$ fraction of the bins are marked at the insertion time of any ball.
- A ball is called marked if all of its $d$ locations points to a marked bin. Let $B$ denote the number of marked bins.

Lemma: $B \leq 2 \max\{n \alpha^d, \ln^2 n\}$, w.h.p.

Proof: $B$ is a sum of independent indicator variables with mean at most $\alpha^d$. Applying a Chernoff bound with $\mu = \max\{n \alpha^d, \ln^2 n\}$ gives

$$\Pr[B \geq 2\mu] \leq e^{-\mu/3} \leq e^{-\ln^2 n/3} \leq n^{-\ln n/3} \leq n^{-\kappa},$$

for every constant $\kappa > 0$, provided that $n$ is sufficiently large. □
The lemma gives \( n \alpha_i \leq B_i \leq \max\{2n \alpha_{i-1}^d, 2 \ln^2 n\} \), w.h.p.

Repeated application (for at most \( n \) times), yields the following recurrence (holding w.h.p.)

\[
\alpha_i \leq \max\left\{2\alpha_{i-1}^d, \frac{2 \ln^2 n}{n}\right\}, \text{ for } i \geq 4,
\]

\[
\alpha_3 \leq \frac{1}{3}.
\]

A simple induction shows

\[
\alpha_i \leq \max\left\{2\sum_{j=0}^{i-4} d^j \left(\frac{1}{3}\right)^{d^{i-3}}, \frac{2 \ln^2 n}{n}\right\} \leq \max\left\{\left(\frac{2}{3}\right)^{d^{i-3}}, \frac{2 \ln^2 n}{n}\right\}.
\]

Hence, for \( \ell \geq \log_d \log_{3/2} n + 3 \), we have \( \alpha_\ell \leq \frac{2 \ln^2 n}{n} \), w.h.p.
• The probability that \( k \) or more balls have height larger than \( \ell \) is at most
\[
\binom{n}{k} \left(\frac{2 \ln^2 n}{n}\right)^{dk} \leq \left(\frac{en}{k}\right)^k \left(\frac{2 \ln^2 n}{n}\right)^{dk} \leq \left(\frac{e(2 \ln^2 n)^d}{n^{d-1/k}}\right)^k
\]
because each of the \( dk \) locations of these balls have to point to the at most \( 2 \ln^2 n \) bins with at least \( \ell \) balls.

• For \( k \geq 0, d \geq 2 \) and sufficiently large \( n \), it holds \( n^{d-1.5} \geq e(2 \ln^2 n)^d / k \).
  In this case, the above probability is at most \( n^{-k/2} \).

• Consequently, the number of balls with height \( > \ell \) is \( O(1) \), w.h.p.

• Thus, w.h.p., the maximum load is
\[
L \leq \ell + O(1) \leq \log_d \log_{3/2} n + 3 + O(1) = \log_d \log n + O(1) .
\]

\(\Box\)
Algorithm ALWAYS-GO-LEFT

- partition set of bins into $d \geq 2$ groups of same size
- choose one alternative from each group at random
  - give ball to alternative with smallest load
  - in case of a tie, ALWAYS-GO-LEFT

Claim: Unfairness yields a much better load balancing
Why unfair tie breaking?

- Consider insertion of a ball $b$.
- Assume $\alpha(b) = \frac{1}{2}(\alpha^{(\text{left})} + \alpha^{(\text{right})})$ is fixed.

- The probability that $b$ is inserted at height $\geq i$ is

$$p_i(b) = \alpha^{(\text{left})} \cdot \alpha^{(\text{right})}$$

- This term is maximized for $\alpha^{(\text{left})} = \alpha^{(\text{right})} !!!$
Theorem - upper bound

Suppose \( n \) balls are inserted into \( n \) bins using \( d \)-left. Then

\[
L = \frac{\ln \ln n}{d \ln \phi_d} + O(1) \text{ w.h.p.}
\]

with \( 1.61 < \phi_2 < \phi_3 < \cdots < 2 \).

\( \phi_d \) is a generalization of the golden ratio, that is, we define \( \phi_d \) such that

\[
\lim_{k \to \infty} \frac{F_d(k)}{\phi_d^k} = 1,
\]

where \( F_d(k) \) is the \( k \)-th \( d \)-ary Fibonacci number.

Improvement for large \( d \) is obvious. But even for \( d = 2 \),

\[
\frac{\ln \ln n}{d \ln \phi_d} < 0.7 \frac{\ln \ln n}{\ln d}.
\]
The Proof

- Fix a ball $b$. Suppose this ball is placed in group $j$ and has height $i$.

Then, before the insertion of $b$,
- alternatives 0 to $j - 1$ have load at least $i$
- alternative $j$ has load $i - 1$
- alternatives $j + 1$ to $d - 1$ have load at least $i - 1$
• $\beta_h := \text{fraction of bins with load } \geq \frac{h}{d} \text{ in group } h \mod d$.

- A ball that ends up in layer $h$ (or above) has its $d$ locations pointing to the layers $h-1, \ldots, h-d$, resp.

- We obtain

\[
\mathbb{E} [\beta_h] \leq d \prod_{k=1}^{d} \beta_{h-k}.
\]

(The factor $d$ is required because the number of balls is $n$ whereas the number bins per group is only $n/d$.)
Using Chernoff bounds, we obtain the following recurrence holding w.h.p.

\[
\begin{align*}
\beta_h & \leq 2 \max \left\{ \prod_{k=1}^{d} \beta_{h-k}, \frac{\log^2 n}{n} \right\} \quad \text{for } h \geq 4d \\
\beta_h & \leq d \left( \frac{1}{3} \right)^d \quad \text{for } h \in \{3d, \ldots, 3d + d - 1\}
\end{align*}
\]

Now, one can show by induction, unless \( \beta_h \leq 2 \frac{\log^2 n}{n} \),

\[
\begin{align*}
\beta_h & \leq (2d)^{\sum_{j=1}^{h+1} - 4d} F_d(j) \left( d \left( \frac{1}{3} \right)^d \right)^{F_d(h-4d+3)} \\
& \leq \left( 2d^2 \left( \frac{1}{3} \right)^d \right)^{F_d(h-4d+3)} \leq \left( \frac{8}{9} \right)^{F_d(h-4d+3)} = c^{-\phi_d^{h-4d+3}}
\end{align*}
\]

for a suitable positive constant \( c < 1 \).
• We have $\beta_h = c^{-\phi_d h - 4d + 3}$, w.h.p.

• Observe $\alpha_i = \frac{1}{d} \sum_{j=0}^{d-1} \beta_{(i-1)d+j} \leq c^{-\phi_d (i-1)d - 4d + 3} = c^{-\phi_d (i-5)d + 3}$.

• Analogously to the symmetric allocation scheme, it follows (basically by setting $\alpha_L \approx \frac{1}{n}$) that

$$L = \frac{\log \phi_d \log_c n}{d} + O(1) = \frac{\ln \ln n}{d \ln \phi_d} + O(1).$$
Theorem - lower bound

- Let $\text{choose}(n, d)$ denote any function that selects $d$ out of $n$ bins at random. These $d$ locations may be chosen
  - non-uniformly, and
  - dependent among each other.

- Let $\mathcal{A}$ be any on-line algorithm that, for every ball,
  - selects $d$ alternatives using $\text{choose}(n, d)$ and
  - places the ball into one of these alternatives.

- Suppose $n$ balls are placed into $n$ bins using $\mathcal{A}$. Then

$$L \geq \frac{\ln \ln n}{d \ln \phi_d} - O(1) \text{ w.h.p.}$$
## Experimental Results

<table>
<thead>
<tr>
<th>$n$</th>
<th>$d = 1$</th>
<th>$d = 2$</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>single-choice</td>
<td>uniform</td>
<td>go-left</td>
</tr>
<tr>
<td>$2^8$</td>
<td>3 \ldots 1%</td>
<td>2 \ldots 10%</td>
<td>2 \ldots 29%</td>
</tr>
<tr>
<td></td>
<td>4 \ldots 40%</td>
<td>3 \ldots 90%</td>
<td>3 \ldots 71%</td>
</tr>
<tr>
<td></td>
<td>5 \ldots 41%</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>6 \ldots 15%</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>7 \ldots 3%</td>
<td></td>
<td></td>
</tr>
<tr>
<td>$2^{16}$</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>7 \ldots 48%</td>
<td>3 \ldots 64%</td>
<td>3 \ldots 100%</td>
</tr>
<tr>
<td></td>
<td>8 \ldots 43%</td>
<td>4 \ldots 36%</td>
<td></td>
</tr>
<tr>
<td></td>
<td>9 \ldots 9%</td>
<td></td>
<td></td>
</tr>
<tr>
<td>$2^{24}$</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>9 \ldots 12%</td>
<td>4 \ldots 100%</td>
<td>3 \ldots 44%</td>
</tr>
<tr>
<td></td>
<td>10 \ldots 73%</td>
<td></td>
<td>4 \ldots 56%</td>
</tr>
<tr>
<td></td>
<td>11 \ldots 13%</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>12 \ldots 2%</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
## Experimental Results

<table>
<thead>
<tr>
<th>$n$</th>
<th>$d = 2$</th>
<th></th>
<th>$d = 4$</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>uniform</td>
<td>go-left</td>
<td>uniform</td>
<td>go-left</td>
</tr>
<tr>
<td>$2^8$</td>
<td>2 ... 10%</td>
<td>2 ... 29%</td>
<td>2 ... 99%</td>
<td>2 ... 100%</td>
</tr>
<tr>
<td></td>
<td>3 ... 90%</td>
<td>3 ... 71%</td>
<td>3 ... 1%</td>
<td></td>
</tr>
<tr>
<td>$2^{16}$</td>
<td>3 ... 64%</td>
<td>3 ... 100%</td>
<td>2 ... 23%</td>
<td>2 ... 100%</td>
</tr>
<tr>
<td></td>
<td>4 ... 36%</td>
<td></td>
<td>3 ... 77%</td>
<td></td>
</tr>
<tr>
<td>$2^{24}$</td>
<td>4 ... 100%</td>
<td>3 ... 44%</td>
<td>3 ... 100%</td>
<td>2 ... 100%</td>
</tr>
<tr>
<td></td>
<td></td>
<td>4 ... 56%</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>