Randomized Load Balancing: The Power of 2 Choices

Ioannis Panageas

June 3, 2010

Balls and Bins

Problem

We have m balls that are thrown into n bins, the location of each ball chosen independently and uniformly at random.

Balls and Bins

Problem

We have m balls that are thrown into n bins, the location of each ball chosen independently and uniformly at random.

- How many bins are empty?

Balls and Bins

Problem

We have m balls that are thrown into n bins, the location of each ball chosen independently and uniformly at random.

- How many bins are empty?
- What is the expected maximum load?

Balls and Bins

Problem

We have m balls that are thrown into n bins, the location of each ball chosen independently and uniformly at random.

- How many bins are empty?
- What is the expected maximum load?
  - Recall from birthday paradox we have that for $m = \Omega(\sqrt{n})$ at least one of the bins has more than one ball with probability $\geq \frac{1}{2}$

Balls and Bins

Problem

We have m balls that are thrown into n bins, the location of each ball chosen independently and uniformly at random.

- How many bins are empty?
- What is the expected maximum load?
  - Recall from birthday paradox we have that for $m = \Omega(\sqrt{n})$ at least one of the bins has more than one ball with probability $\geq \frac{1}{2}$

Proof:

- $P(E'_1 \cup ... \cup E'_k) \leq \sum_{j=1}^{k} P(E'_j) = \sum_{j=1}^{k} \frac{j-1}{n} = \frac{k(k-1)}{2n} < \frac{k^2}{2n}$

Lemma

The probability p such that the maximum load is more than $3\frac{\ln n}{\ln\ln n}$ is at most $\frac{1}{n}$ for n sufficiently large.

Proof:

- Using union bound we have that $p \leq n\binom{n}{M}(\frac{1}{n})^M \leq \frac{n}{M!} \leq n(\frac{e}{M})^M$
- The function $(\frac{e}{M})^M$ is decreasing so for $M \geq 3\frac{\ln n}{\ln\ln n}$ follows that $n(\frac{e}{M})^M \leq \frac{1}{n}$.

Theorem

When n balls are thrown independently and uniformly at random into n bins, the maximum load is at least $\frac{\ln n}{\ln\ln n}$ with probability $p \geq 1 - \frac{1}{n}$ for n sufficiently large.

The main difficulty in analyzing balls-in-bins problem and proving the above lemma is handling the dependencies.

Theorem

When n balls are thrown independently and uniformly at random into n bins, the maximum load is at least $\frac{\ln n}{\ln\ln n}$ with probability $p \geq 1 - \frac{1}{n}$ for n sufficiently large.

The main difficulty in analyzing balls-in-bins problem and proving the above lemma is handling the dependencies.

- So the solution is using Poisson distribution!

Lemma 1.

Let $X_i^{(m)}$ be the number of balls in i-th bin and $Y_i^{(m)}$ be independent Poisson random variables with mean $\frac{m}{n}$. The distribution of $(Y_1^{(m)}, ..., Y_n^{(m)})$ conditioned on $\sum_i Y_i^{(m)} = k$ is the same as $(X_1^{(k)}, ..., X_n^{(k)})$, regardless the value of m.

Lemma 1.

Let $X_i^{(m)}$ be the number of balls in i-th bin and $Y_i^{(m)}$ be independent Poisson random variables with mean $\frac{m}{n}$. The distribution of $(Y_1^{(m)}, ..., Y_n^{(m)})$ conditioned on $\sum_i Y_i^{(m)} = k$ is the same as $(X_1^{(k)}, ..., X_n^{(k)})$, regardless the value of m.

Proof:

- Throwing k balls into n bins , the probability $(X_1, ..., X_n) = (k_1, ..., k_n)$ such that $\sum_i k_i = k$ is
$$\frac{\binom{k}{k_1}\binom{k-k_1}{k_2}\ldots\binom{k-k_1-\ldots-k_{n-1}}{k_n}}{n^k} = \frac{k!}{k_1!k_2!\ldots k_n!n^k}$$
- The probability $(Y_1^{(m)}, ..., Y_n^{(m)}) = (k_1, ..., k_n)$ such that $\sum_i Y_i^{(m)} = k$ is
$$\frac{P((Y_1^{(m)}=k_1)\cap\ldots\cap(Y_n^{(m)}=k_n))}{P(\sum_i Y_i^{(m)}=k)} = \frac{e^{-m/n}(m/n)^{k_1}\ldots e^{-m/n}(m/n)^{k_n}k!}{k_1!\ldots k_n!e^{-m}m^k}$$

Lemma 2.

Let $f(x_1, ..., x_n)$ be a nonnegative function. Then $E[f(X_1^{(m)}, ..., X_n^{(m)})] \leq e\sqrt{m}E[f(Y_1^{(m)}, ..., Y_n^{(m)})]$.

Lemma 2.

Let $f(x_1, ..., x_n)$ be a nonnegative function. Then $E[f(X_1^{(m)}, ..., X_n^{(m)})] \leq e\sqrt{m}E[f(Y_1^{(m)}, ..., Y_n^{(m)})]$.

Proof:
$E[f(Y_1^{(m)}, ..., Y_n^{(m)})] = \sum_{t=0}^{\infty} E[f(Y_1^{(m)}, ..., Y_n^{(m)})| \sum_{i=1}^{n} Y_i^{(m)} = t]$
$\cdot Pr(\sum_{i=1}^{n} Y_i^{(m)} = t) \geq E[f(Y_1^{(m)}, ..., Y_n^{(m)})| \sum_{i=1}^{n} Y_i^{(m)} = m]$
$\cdot Pr(\sum_{i=1}^{n} Y_i^{(m)} = m)$
$= E[f(X_1^{(m)}, ..., X_n^{(m)}) \frac{m^m e^{-m}}{m!} \geq E[f(X_1^{(m)}, ..., X_n^{(m)})$
$\frac{m^m e^{-m}}{m!} \geq E[f(X_1^{(m)}, ..., X_n^{(m)}) \frac{1}{e\sqrt{m}}$.

Proof of Theorem:
With m=n the probability in the Poisson case that a fixed bin has at least M balls is at least $\frac{1}{eM!}$. So because the bins are independent we have that the probability no bin has at least M balls is at most $(1-\frac{1}{eM!})^n \leq e^{\frac{-n}{eM!}}$. Thus by choosing M such that $e^{\frac{-n}{eM!}} \leq \frac{1}{n^2}$ we have that from the previous lemma that the fact in the exact case has probability at most $e\frac{\sqrt{n}}{n^2} < \frac{1}{n}$. The (largest) choice is $M = \frac{\ln n}{\ln \ln n}$

Balls and Bins

Problem

Each ball comes with d possible destination bins, each chosen independently and uniformly at random and is placed in the least full bin among the d locations (ties broken randomly).

Balls and Bins

Problem

Each ball comes with d possible destination bins, each chosen independently and uniformly at random and is placed in the least full bin among the d locations (ties broken randomly).

- What is the expected maximum load now?

Balls and Bins

Problem

Each ball comes with d possible destination bins, each chosen independently and uniformly at random and is placed in the least full bin among the d locations (ties broken randomly).

- What is the expected maximum load now?
  - We are going to prove that the maximum load decreases exponentially...

Theorem

The maximum load for the problem above is at most $\frac{\ln\ln n}{\ln d} + O(1)$ with probability $1 - o(\frac{1}{n})$

Proof Sketch:

Theorem

The maximum load for the problem above is at most $\frac{\ln\ln n}{\ln d} + O(1)$ with probability $1 - o(\frac{1}{n})$

Proof Sketch:

- We wish to find a sequence of values b_i such that the number of bins with load at least i is bounded above by b_i w.h.p. Suppose we know b_i, we want to find b_{i+1}. If a ball has height at least $i+1$ only if each of its d choices for a bin has load at least i. Therefore the probability that a ball has height at least $i+1$ is at most $(\frac{b_i}{n})^d$.

Theorem

The maximum load for the problem above is at most $\frac{\ln\ln n}{\ln d} + O(1)$ with probability $1 - o(\frac{1}{n})$

Proof Sketch:

- We wish to find a sequence of values b_i such that the number of bins with load at least i is bounded above by b_i w.h.p. Suppose we know b_i, we want to find b_{i+1}. If a ball has height at least $i+1$ only if each of its d choices for a bin has load at least i. Therefore the probability that a ball has height at least $i+1$ is at most $(\frac{b_i}{n})^d$.
- Using standard bounds on Bernoulli trials, it follows that $b_{i+1} \leq cn(\frac{b_i}{n})^d$ for constant c, so by selecting $j = O(\ln\ln n)$ we are done ($b_j < 1$).

Lemma 1

$$P(B(n,p) \geq enp) \leq e^{-np}. \quad (1)$$

Lemma 1

$P(B(n, p) \geq enp) \leq e^{-np}$. (1)

- Using Chernoff Bounds $P(X \geq (\delta + 1)\mu) \leq (\frac{e^{\delta}}{(1+\delta)^{(1+\delta)}})^{\mu}$ for $\delta = e - 1$ we conclude the proof.

Lemma 1

$P(B(n,p) \geq enp) \leq e^{-np}$. (1)

- Using Chernoff Bounds $P(X \geq (\delta+1)\mu) \leq (\frac{e^{\delta}}{(1+\delta)^{(1+\delta)}})^{\mu}$ for $\delta = e - 1$ we conclude the proof.

Lemma 2

Let $X_1, X_2, ..., X_n$ be a sequence of r.v in an arbitrary domain, and let $Y_1, Y_2, ..., Y_n$ be a sequence of binary r.v such that $Y_i = Y_i(X_1, ..., X_{i-1})$. If $P(Y_i = 1 | X_1, ..., X_{i-1}) \leq (\text{or} \geq)p$ then $P(\sum_{i=1}^{n} Y_i \geq k) \leq (\text{or} \geq) P(B(n,p) \geq k)$

Y_i is less (or more) likely to take value 1 than an independent Bernoulli trial.

Proof of theorem:

Proof of theorem:

- Let h(t) be the height of the t-th ball that is placed , $v_i(t)$ the number of bins with load at least i and $\mu_i(t)$ the number of balls with height at least i at time t. Suppose $\beta_6 = \frac{n}{2e}$, $\beta_{i+1} = ne(\frac{\beta_i}{n})^d$ and E_i be the event that $v_i(n) \leq \beta_i$. We want to find the largest i , such that if E_i holds then E_{i+1} holds w.h.p.

Proof of theorem:

- Let $h(t)$ be the height of the t-th ball that is placed , $v_i(t)$ the number of bins with load at least i and $\mu_i(t)$ the number of balls with height at least i at time t. Suppose $\beta_6 = \frac{n}{2e}$, $\beta_{i+1} = ne(\frac{\beta_i}{n})^d$ and E_i be the event that $v_i(n) \leq \beta_i$. We want to find the largest i , such that if E_i holds then E_{i+1} holds w.h.p.
- Fix i and assume $Y_t = 1$ iff $h(t) \geq i+1$ and $v_i(t-1) \leq \beta_i$. So $P(Y_t = 1 | \omega_1, ..., \omega_{t-1}) \leq (\frac{\beta_i}{n})^d = p_i$.

Proof of theorem:

- Let $h(t)$ be the height of the t-th ball that is placed , $v_i(t)$ the number of bins with load at least i and $\mu_i(t)$ the number of balls with height at least i at time t. Suppose $\beta_6 = \frac{n}{2e}$, $\beta_{i+1} = ne(\frac{\beta_i}{n})^d$ and E_i be the event that $v_i(n) \leq \beta_i$. We want to find the largest i , such that if E_i holds then E_{i+1} holds w.h.p.
- Fix i and assume $Y_t = 1$ iff $h(t) \geq i + 1$ and $v_i(t-1) \leq \beta_i$. So $P(Y_t = 1 | \omega_1, ..., \omega_{t-1}) \leq (\frac{\beta_i}{n})^d = p_i$.
- Conditioned on E_i we have $\sum Y_t = \mu_{i+1}(n)$. Thus $P(v_{i+1} \geq \beta_{i+1} | E_i) \leq P(\mu_{i+1} \geq \beta_{i+1} | E_i) \leq \frac{P(\sum Y_t \geq \beta_{i+1})}{P(E_i)}$ $\leq \frac{P(B(n,p_i) \geq \beta_{i+1})}{P(E_i)} \leq \frac{1}{e^{np_i} P(E_i)}$ (Lemma 2 - Lemma 1).

- Case $p_i n \geq 2 \ln n \Rightarrow P(\neg E_{i+1} | E_i) \leq \frac{1}{n^2 P(E_i)}$.

- Case $p_i n \geq 2\ln n \Rightarrow P(\neg E_{i+1}|E_i) \leq \frac{1}{n^2 P(E_i)}$.
- Case $p_i n < 2\ln n$: Let i^* be the smallest value such that $p_i n < 2\ln n$.
  Then $i^* \leq \frac{\ln\ln n}{\ln d} + O(1)$ (we can prove it using induction to prove $\beta_{i+6} \leq n/2^{d^i}$). Finally we have to prove that $P(\mu_{i^*+2} \geq 1) = O(\frac{1}{n})$. Using the fact that $P(v_{i^*+1} \geq 6\ln n|E_{i^*}) \leq P(B(n, 2\ln n/n) \geq \ln n) \leq \frac{1}{n^2 P(E_{i^*})}$ and $P(\mu_{i^*+2} \geq 1|\mu_{i^*+1} \leq 6\ln n) \leq \frac{P(B(n,((6\ln n)/n)^d))}{P(\mu_{i^*+1} \leq 6\ln n)}$ $\leq \frac{n(6\ln n)/n)^d}{P(\mu_{i^*+1})}$ it follows that $P(\mu_{i^*+2} \geq 1) = O(\frac{1}{n})$.

Using the same technique, we can prove the following theorem for the lower bound.

Theorem

The maximum load for the problem above is at least $\frac{\ln\ln n}{\ln d} - O(1)$ with probability $1 - o(\frac{1}{n})$.

There are also other known techniques for proving the theorems above concerning the Upper and Lower Bound.

- Witness tree method (tree of events). probability of occurence of bad events bounded above by probability of occurence of witness tree.
- Fluid limit models (describing the system by differential equations).

Here we mention another mechanism of replacing each ball into a bin.

- Always-Go-Left Algorithm:

Here we mention another mechanism of replacing each ball into a bin.

- Always-Go-Left Algorithm:
    - Partition the bins into d groups of almost equal size ($\Theta(\frac{n}{d})$).

Here we mention another mechanism of replacing each ball into a bin.

- Always-Go-Left Algorithm:
  - Partition the bins into d groups of almost equal size $(\Theta(\frac{n}{d}))$.
  - For each ball choose one location from each group.

Here we mention another mechanism of replacing each ball into a bin.

- Always-Go-Left Algorithm:
  - Partition the bins into d groups of almost equal size ($\Theta(\frac{n}{d})$).
  - For each ball choose one location from each group.
  - The ball is placed in the bin with the minimum load (ties are broken by inserting the ball to the leftmost bin).

Here we mention another mechanism of replacing each ball into a bin.

- Always-Go-Left Algorithm:
  - Partition the bins into d groups of almost equal size $(\Theta(\frac{n}{d}))$.
  - For each ball choose one location from each group.
  - The ball is placed in the bin with the minimum load (ties are broken by inserting the ball to the leftmost bin).

  It's proven that the maximum load is $\frac{\ln\ln n}{d \ln \phi_d} + O(1)$ w.h.p where $\phi_d = \lim_{k\to\infty} \sqrt[k]{F_d(k)}$. As the limit converges we have that this algorithm has better load balancing than the previous one.

Bucket-sort

Suppose we have $n = 2^m$ elements to be sorted, each one chosen independently and uniformly at random from a range $[0, 2^k)$ with $k \geq m$ and n buckets. The algorithm has two stages:

Bucket-sort

Suppose we have $n = 2^m$ elements to be sorted, each one chosen independently and uniformly at random from a range $[0, 2^k)$ with $k \geq m$ and n buckets. The algorithm has two stages:

- We place the numbers in the buckets (O(n) ?). The j-th bucket contains the numbers whose first m binary digits make j.

Bucket-sort

Suppose we have $n = 2^m$ elements to be sorted, each one chosen independently and uniformly at random from a range $[0, 2^k)$ with $k \geq m$ and n buckets. The algorithm has two stages:

- We place the numbers in the buckets (O(n) ?). The j-th bucket contains the numbers whose first m binary digits make j.
- We sort each bucket using bubblesort (doesn't matter which sorting algorithm we choose) and concatenate the sorted buckets.

Bucket-sort

Suppose we have $n = 2^m$ elements to be sorted, each one chosen independently and uniformly at random from a range $[0, 2^k)$ with $k \geq m$ and n buckets. The algorithm has two stages:

- We place the numbers in the buckets ($O(n)$?). The j-th bucket contains the numbers whose first m binary digits make j.
- We sort each bucket using bubblesort (doesn't matter which sorting algorithm we choose) and concatenate the sorted buckets.

The algorithm runs in $O(n)$: Suppose X_j be a random variable representing the number of integers in j-th bucket. Then X_j follows binomial distribution $B(n, \frac{1}{n})$ so to find the complexity of the algorithm we have to compute $\sum_j E[X_j^2]$. Using the fact that $E[X^2] = Var[X] + E[X]^2 = np(1-p) + (np)^2 = 2 - \frac{1}{n}$ we have that the algorithm runs in $O(n)$.

Chain Hashing

Suppose we want to make a password checker, which prevents people from using unacceptable passwords. We have a set S of m words and we want to check for a given a word, if it belongs or not to S. One easy and well-known idea is binary search in O(logm) if we have S saved as a sorted array. Another approch is using a random hash function and a hash table of size n. We make the assumption that for a given x , $P(f(x) = j) = \frac{1}{n}$, f(x) is fixed and that the values of f are independent. Now the complexity of the algorithm(expected worst case) is the maximum load of a bin....

Chain Hashing

Suppose we want to make a password checker, which prevents people from using unacceptable passwords. We have a set S of m words and we want to check for a given a word, if it belongs or not to S. One easy and well-known idea is binary search in O(logm) if we have S saved as a sorted array. Another approch is using a random hash function and a hash table of size n. We make the assumption that for a given x , $\mathrm{P}(\mathrm{f}(\mathrm{x}) = \mathrm{j}) = \frac{1}{\mathrm{n}}$, f(x) is fixed and that the values of f are independent. Now the complexity of the algorithm(expected worst case) is the maximum load of a bin.... $\mathrm{O}(\frac{\mathrm{lnn}}{\mathrm{lnlnn}})$

Chain Hashing

Another approach for hashing in order to balance the load is the use of two hash functions (d = 2). The two hash functions define two possible entries in the hash table for each item. The item is inserted to the location that is least full. So w.h.p the maximum time to find an item is (O(lnlnn)). However this improvement leads to double the average search time because we look at two bucket instead of one.

The balls-and-bins method can be used in different fields of cs.

- Operating systems (Dynamic resource allocation)
- Game Theory (Congestion Games - Routing)
- Queueing Theory
- DataBases
- other