Randomized Algorithm Tutorial

Nov. 27 2008

Your TA: Abner C.Y. Huang
BarrosH@gmail.com
Ball and Bin Model

- Simple Model
- Concrete Model
Applications

- Randomized load balancing
- Data allocation (flash)
- Hashing
- Routing
Maximum Load (Revisited)

Lemma: When $n$ balls are thrown to $n$ bins, independently and uniformly at random, the maximum load is at least $\ln n/\ln \ln n$ with high probability (at least $1-1/n$)

- Roughly, we have the maximum load $O(\ln n/\ln \ln n)$
Can we do this better?
Idea: multiple-choices allocation

- choose a small sample of bins at random
- inspect bins in and place ball into one of them with fewer number of balls
The Power of Two Choices

**Theorem**: for every ball, choosing $d$ alternatives uniformly at random, the maximum load is

$$O(\ln \ln n / \ln d)$$

with high probability.
Cuckoo hashing

- Multiple-Way hashing.
- The new key is inserted in one of its two possible locations, "kicking out", that is, displacing any key that might already reside in this location.
- A simple and practical scheme with worst case constant lookup time.
- Cuckoo hashing is invented at 2001, Bloom filter is invented at 1970.
Cuckoo Hashing Examples

![Diagram of Cuckoo Hashing Examples]
Cuckoo Hashing Examples
Cuckoo Hashing Examples
Cuckoo Hashing Examples
Cuckoo Hashing Examples
Cuckoo Hashing Properties

- *Worst case constant lookup time.*
- Simple to build, design.
- Lookups using two probes (optimal).
- Efficient in the average case.

- However, it needs some theoretical assumptions.
The Power of Two Choices

**Theorem**: for every ball, choosing $d$ alternatives uniformly at random, the maximum load is

$$O(\ln \ln n / \ln d)$$

with high probability.
Let's try to prove this

• Challenges:
  – This proof is not so difficult in technical detail.
  – However, there are a lot of magic number.
  – And it adapts circuitous approach.
- $h(t)$: the height of a ball. The height $h(t)$ of a ball $t$ means the ball $t$ is the $h(t)$-th ball thrown into the bin.
- $v_i(t)$: the number of balls with height at least $i$ after throwing the $t$-th ball.
- $u_i(t)$: the number of bins with at least $i$ balls after throwing the $t$-th ball.
- Observe that \( \forall t, u_i(t) \leq v_i(t) \).
- Consider throwing \( n \) balls into \( n \) bins, we want to bound \( u_i(n) \forall i \).
- We can get a trivial bound.
Consider the Two-Choices method.

Consider $b_i$ as another bound for $v_i(n)$, i.e.,
$v_i(t) \leq v_i(n) \leq b_i$.

When we threw $t$-th ball, the case that $h(t) = i + 1$ occurs only if both two picked bins have $i$ balls. The probability of this case is
\[
\frac{b_i}{n} \frac{b_i - 1}{n} \sim \left( \frac{b_i}{n} \right)^2.
\]

In general, for $d$-choices method, the probability $p_i$ of this case is at most \( \left( \frac{b_i}{n} \right)^d \).
If we look this process as a binomial random variable $B(n, p_i)$ where each Bernoulli trail is defined by $Pr(X_j = i + 1) = p_i$, then we can use Chernoff bound to realize the bound $b_i$.

$E[B(n, p_i)] = np_i = n \left(\frac{b_i}{n}\right)^d$.

By Chernoff bound, we have

$Pr(B(n, p_i) \geq 2np_i) \leq e^{-np_i/3}$.

Hence we have a bound $b_{i+1} \sim 2np_i = 2n \left(\frac{b_i}{n}\right)^d$ with high probability.
Let $b_4 = \frac{n}{4}$.

$b_{i+1} \sim 2n \left(\frac{b_i}{n}\right)^d$ is an recurrence relation indeed. By solving this, we can get the formula.

$$b_{i+4} \leq \frac{n}{2^{d^i}}$$

Thus one might guess that the maximum load is $g = \frac{\ln \ln n}{\ln d}$ as $\frac{b_g}{n} \sim \frac{1}{n}$.

Note that we might derive difference bounds $f_i$ by using larger derivation in Chernoff bound.
However, $b_i$ is an approximated bound, we can’t guarantee that $v_i(n) \leq b_i$ always.

We have $Pr(v_4(n) \leq b_4) = Pr(v_4(n) \leq \frac{n}{4}) = 1$.

If we defined an event $E_i$ for that $(v_i(n) \leq b_i)$ holds, what the value $i^*$ is such that those events start to fail? Does it provide good bound? How to estimate it?
An idea is to guess a value to estimate $i^\ast$.

Let’s pick $i^\ast$ as the smallest value such that $b_{i^\ast} < 12 \ln n$, i.e., we guess that $E_i$ doesn’t hold when $b_i$ become too small.

And we hope this is also bounded with high probability.
By Chernoff bound, we have

\[
\Pr(B(n, p_i) > b_{i+1} \mid E_i) \leq \Pr(B(n, p_i) > 2np_i \mid E_i) \leq \frac{1}{e^{np_i/3}\Pr(E_i)}
\]

Since we want to bound this with high probability, we should choose the proper \( i \).

Since \( np_i = 6 \ln n \), we can bound it with \( O\left(\frac{1}{n^2\Pr(E_i)}\right) \).
Now we try to derive the value \( i^* \). Since 
\[
b_{i+1} = 2np_i,
\]
we have
\[
p_{i^*} = \left( \frac{b_{(i^* - 4) + 4}}{n} \right)^d \leq \frac{1}{2^{d_{i^* - 3}}} \leq \frac{6 \ln n}{n}
\]

Since \( \frac{1}{n} \leq \frac{6 \ln n}{n} \), by solving \( \frac{1}{2^{d_{i^* - 3}}} \leq \frac{1}{n} \) we have
\[
i^* = \frac{\ln \ln n}{\ln d} + O(1) = o(n).
\]
• We will skip the details here . . . .

• However, if we can prove $O\left(\frac{1}{n^2 \Pr(E_i)}\right)$ is small enough, then eventually we will derive the bound that $\Pr(\nu_{i*} > 1) = o\left(\frac{1}{n}\right)$.

• The details please refer to the textbook.
If you are interested in...

we can do it even better

\[
\frac{\ln \ln n}{d \ln \phi_d}
\]
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
THANK YOU