Lecture Notes: Power of two choices.

Overview: motivation.

Consider a set of n servers. Consider the following possible strategies for assigning a stream of jobs to the servers which previous jobs finished.

- As jobs come in choose a server uniformly at random.
- As jobs come in choose two servers and pick the one that is least loaded.
- As jobs come in you can assign each to the least loaded server.

The last strategy is of course the best in terms of balancing the load, but you also need to continually measure and order the loads on all the servers.

Today, we analyze the first two situations in a simple context.

Balls in Bins, yet again

Recall the following situation.

Experiment: Toss $n/8$ labeled balls into n bins.

(Recall that when you actually did a similar experiment where you placed 1000 balls uniformly at random in 1000 vbins that the maximum number of balls in any bin hovered around 9.)

Fact: The maximum number of balls in any bin is $O(\log n/\log\log n)$ with probability at least $1/n^c$ for some constant c.

Proof: Upper bound the probability that a bin contains more than k balls in it by

$$\binom{n/8}{k}\left(\frac{1}{n}\right)^k \leq \left(\frac{e}{8k}\right)^k < \left(\frac{1}{k}\right)^k.$$

(Why?)

The probability that *any* bin contains more than k balls is at most n times the quantity above. Thus, the probability that *any* bin contains more than $k = c' \log n/\log\log n$ balls is at most $1/n^{c'-1}$. $\square$

Now consider the following situation.

Experiment 2: For each $n/8$ balls, choose two of n bins uniformly at random and place the ball into the bin with the fewest balls.

(Recall that when you actually did a similar experiment where you placed 1000 balls with two random choices in 1000 bins that the maximum number of balls in any bin hovered around 4.)

Today, we will prove the following theorem.

Theorem 1 *With high probability, the maximum number of balls on any bin is* $O(\log\log n)$.

Note that this is significantly lower than for the first experiment.

First, we define the following random graph that is generated by the random choices in our experiment. Let $G = (V, E)$, where

$$V = \{1, ..., n\}$$

and

$$E = \{e_i = (u, v) : \text{ ball } i \text{ chose bin } u \text{ and bin } v\}.$$

Here are some questions about this graph.

How many singletons? (Unimportant for remainder of lecture.) Answer: $n(1 - 1/n)^{2n/8} \rightarrow ne^{1/4}$.

What is the maximum degree? Answer: $\Theta(\log n / \log\log n)$. Why?

What is the average degree? Answer: $1/4$. Why?

Two very critical properties for this proof. We will prove these later.

Lemma 2 *With high probability the maximum size of any component is* $\Theta(\log n)$.

Lemma 3 *With high probability the average induced degree on a subset of nodes, S, is at most 5. (The average induced degree on a subset of nodes S is the average degree of the nodes in the graph induced on S. The induced graph on S is the set of edges both of whose endpoints are in S.)*

Now we examine the following process on the random graph.

Removal Process:

Repeat until no edges.

Remove all edges that are incident to nodes of degree less than 10.

Assuming that theorems 1 and 2 hold for a particular graph, we can prove the following fact.

Fact 1 *The number of iterations required to remove all of the edges is* $O(\log\log n)$ *with high probability.*

Proof: Let k_i be the number of nodes of the largest connected component after iteration i of the removal process. We have by lemma 2 that $k_0 \leq c \log nqqqgq$ with high probability.

Consider S to be set of nodes in a component at any point in the algorithm, the average degree of edges in this component is 5 with high probability. Thus, since remove all the edges incident to nodes with degree less than 10, we must remove at least half of the nodes from the component. (That is, the remaining nodes had degree greater than 10.)

So $k_{i+1} \leq k_i/2$. After $\log k_0 = O(\log \log n)$ iterations the component size becomes 1 and there are no edges left. $\square$

Now we will upper bound the maximum number of balls in a bin by the number of iterations of the removal process to complete the proof.

Fact 2 *Say G corresponds to our experiment, the maximum number of balls in any bin is upperbounded by 10 times the number of iterations of the removal process on G.*

Proof: Given an experiment, we define h_i to be the maximum height that could be assigned to ball i. Note that ball i could be assigned to either of two bins if the number of balls in the bins were equal when ball i is assigned.

We also run the removal process on G, the graph corresponding to the experiment. We define t_i as the iteration in which e_i is removed from G.

We will prove inductively that $h_i \leq 10t_i$.

First, consider a ball i whose edge $e_i = (u, v)$ is removed in the first iteration. Say bin u or v has degree at most 10, i.e., at most 10 balls can be in u. Thus, ball i can have height at msot 10. This follows from the definition of G and e_i. Thus, for those balls, we have $h_i \leq 10$.

Now say a ball i that is removed in iteration t, chooses bins u and v. Assume without loss of generality that u's degree is at most 10 in this iteration. Inductively, we know that the maximum height corresponding to a ball whose edge has been previously removed from u is at most $(t-1)10$.

Since, only 10 more balls will ever be assigned to this bin, the height of bin u is upperbounded by $(t-1)10 + 10 = 10t$. Since ball i could have been assigned to bin u it als must have height at most $10t$.

$\square$

Thus, we know that the maximum height of any ball is $O(\log \log n)$ given lemmas 2 and 3.

We proceed by proving the lemmas.

Proof of lemma 2. We will bound the probability that there is a component of size $k+1$.

The outline of the proof is that given a component of size $k+1$, we will show that some structure existed. Then we argue that the probability that such a structure existed is small. Thus, a component of size $k+1$ is unlikely to exist.

We consider a spanning tree of the component. Then we consider a walk on the tree defined by depth first traversal of the tree. The length of the walk is $2k$. The walk consists of k advancing steps where a new node is first encountered and k retreat steps where we revisit an old node. Thus, the number of such walks is bounded by

$$\binom{2k}{k} n^k \leq (2n)^k .$$

For a particular spanning tree to exist, each of k edges must exist in G. The probability of all of these edges existing is upper bounded by

$$\binom{n/8}{k}\left(\frac{1}{n^2}\right)^k \leq \left(\frac{e}{8kn}\right)^k.$$

Assuming $k > e$, this above is bounded by $(1/8n)^k$.

The probability that any spanning tree exists is at most the number of such spanning trees times the above bound or at most $(1/4)^k$.

This is can be bounded by $1/n^c$ when $k > c\log_4 n$.

$\square$

Proof sketch of lemma 3. You should prove this lemma. Notice that the average induced degree is simply twice the number of edges in S divided by $|S|$. The idea is to upper bound the probalility that more than $5k/2$ edges fall inside a subset of size k, and multiply this by the number of such subsets. $\square$

Summary

light load, you seek?
at two, take a peek

I'd also like to remark that this phenomenon and proof was discovered only recently (1992) by Karp, Luby, and Meyer auf der Heidie, here in Berkeley.