CS 273: January 27th, 2003

Prof. Satish Rao

Load Balancing

We know from last time that a possibility is to just assign each job to some server uniformly at random. Assuming for convenience that we have both n servers and n jobs, we get the following theorem:

Theorem 1

When n balls are thrown into n bins, the maximum number of balls in any bin is $O(\log n)$, and in fact can be shown to be $O\left(\frac{\log n}{\log\log n}\right)$, with high probability.

This is proven by direct calculation:

$$Pr[\text{bin 1 has } \geq \text{ k balls}] \leq \sum_{i=k}^{n} \binom{n}{k} \left(\frac{1}{n}\right)^i \left(1 - \frac{1}{n}\right)^{n-i} \leq \left(\frac{e}{k}\right)^k < \frac{1}{n^2}$$

We can ensure the last inequality is strictly less by choosing a convenient k.

At this point, let's say that "With high probability" means that $Pr[failure] \leq 1 - \frac{1}{n^c}$, where usually we can massage the c by changing constants.

So the consequences for our job/server situation are that in $O\left(\frac{\log n}{\log\log n}\right)$ time, we can run n jobs; but our servers' true capacity is $O\left(\frac{n\log n}{\log\log n}\right)$, so we have an inneficiency of $O\left(\frac{\log\log n}{\log n}\right)$

To overcome this, let's choose a more suitable number of balls to throw into our bins, say m. We know that $\frac{m}{n}$ a lower bound on the maximum load (because it is optimal).

When throwing m balls into n bins:

$$Pr[\text{bin 1 receives } \geq k \text{ balls}] \leq \sum_{i=k}^{n} \binom{m}{k} \left(\frac{1}{n}\right)^i \left(1 - \frac{1}{n}\right)^{m-i} \leq \left(\frac{m}{n}\right)^k \left(\frac{e}{k}\right)^k \leq \left(\frac{me}{nk}\right)^k$$

So to get this expression less than $\frac{1}{n^2}$, we let $k > \frac{2me}{n}$ and also $k > 2\log n$ (this second one follows from $m > n\log n$)

Theorem 2

If $k > \frac{2me}{n}$ and $k > 2\log n$, then the max load is less than k w.h.p.

So in $2e\left(\frac{m}{n}\right)$ time, we can run m jobs, with a much reduced inefficiency of $2e$.

But we keep throwing out the last term of the probability expression, because we know it is < 1. This could be working to our advantage since it is (in this case) $O\left(e^{-m/n}\right)$

The easiest way to get a good result out of that is to abandon the counting technique and use a Chernoff lemma.

Chernoff/Hoeffding Lemmas:

Given n independent and identically distributed zero/one variables $X_1, \cdots, X_n$ such that $X = \sum X_i$ and $E[X] = \mu$.

Lemma 1

$$Pr[X \leq (1-\delta)\mu] \leq e^{-\delta^2\mu/2}$$

Lemma 2

$$Pr[X \geq (1+\delta)\mu] \leq \begin{cases} e^{-\delta^2\mu/3} & 0 \leq \delta \leq 1 \\ e^{-\delta^2\mu/(2+\delta)} & \delta \geq 1 \end{cases}$$

So in our case we have m X_i variables, one per ball. $X_i = \begin{cases} 0 & \text{if ball } i \text{ misses bin 1} \\ 1 & \text{otherwise} \end{cases}$ and $X = \sum X_i =$ load on bin 1

Since we want a small δ, let's use the first form of Lemma 2, and at the end of our calculations, make sure that δ is in the acceptable range.

$$\begin{aligned} Pr[X > (1+\delta)\mu] &\leq e^{-\delta^2\mu/3} \\ \text{We want: } e^{\delta^2\mu/3} &< \frac{1}{n^2} \\ \frac{\delta^2\mu}{3} &< -2\ln n \\ \delta^2 &> \left(\frac{6}{\mu}\right)\ln n = \left(\frac{6n}{m}\right)\ln n \\ \delta &> \sqrt{\left(\frac{6n}{m}\right)\ln n} \\ \rightarrow m \geq 6n\ln n \end{aligned}$$

Theorem 3

If $m > 6n\ln n$ then the max load on any server is $\left(1+\sqrt{\frac{6n\ln n}{m}}\right)\left(\frac{m}{n}\right)$ w.h.p. (in this case, exactly $\frac{1}{n}$)

A New Idea: Two Bins Per Ball

With n jobs and n servers, choose 2 uniformly at random, and choose the least loaded server.

Theorem 4

With this 2-choice strategy, but throwing $\frac{n}{8}$ balls into n bins, the max load on any server is $O(\log\log n)$ w.h.p.

Proof of theorem 4: Define a graph $G = (V, E)$.
$V = \{1, \cdots, n\}$, one for each bin.
$E = \{e_i = (u, v) |$ ball i chooses bins u and $v\}$

There are exactly $\frac{n}{8}$ edges and n vertices, so the average degree is $\frac{1}{4}$. We also know that w.h.p. the max degree of any vertex is $O\left(\frac{\log n}{\log\log n}\right)$, but we really want a better bound than that.

Condition 1

> For any subset S of vertices, the average degree of the induced subgraph on S is ≤ 5 w.h.p.

Condition 2

> The largest connected component has size $O(\log n)$ w.h.p.

Removal Process:
Remove all nodes with degree ≤ 10, and all their incident edges. Repeat until the graph is empty.

Lemma 3

> If Conditions 1 and 2 hold, then the removal process terminates in $O(\log\log n)$ iterations.

Proof of Lemma 3: Consider a connected component. In which iteration is it:w gone? Well, let k_i be its size at iteration i.

By Condition 1, we can conclude that at least $\frac{k_i}{2}$ vertices have degree ≤ 10, and so $\frac{k_i}{2}$ will be left after an iteration. So it takes log(number of nodes in connected component) for the component to be finished.

By Condition 2, we know that the number of nodes is $O(\log n)$, so the total running time is $O(\log\log n)$, and the lemma is proved.

To relate it back to servers, we will use a proof by induction.

Claim:

> The max load on any server is ≤ 10 times the number of iterations of the removal process.

This is because if an edge e_i is removed at iteration t then the height of the corresponding ball in its bin is $h_i \leq 10t$. Once we prove this, it follows that the max load is $O($removal process).

Base Case:
If an edge is removed at the first iteration, then one of the bins that the corresponding ball was looking at had ≤ 10 balls so far (trivial).

Induction:
The maximum height of x balls is $10(t-1)$. So the height of ball $x+1 \leq 10(t-1)+9 \leq 10t$. And we are done.

By the way, one should not that the result of $O(\log\log n)$ is *deterministic* if condition 1 and condition 2 hold.

Proof of Condition 2: A k-node connected component must have $k-1$ edges. Just to clarify the following expression, the first term is choosing k vertices, the second is to choose k balls/edges, and the last is the probability that *both* of the vertices were chosen by that edge.

$$
\begin{aligned}
Pn[\exists k - \text{node connected component}] &\leq \binom{n}{k}\binom{n/8}{k}\left(\frac{k}{n}\right)^{2k} \\
&\leq \left(\frac{ne}{k}\right)^{k}\left(\frac{ne/8}{k}\right)^{k}\left(\frac{k}{n}\right)^{2k} \\
&\leq \left(\frac{e^2}{8}\right)^{k} \\
&\leq \left(\frac{1}{n^2}\right) \text{ (with our choice of nice } k\text{)}
\end{aligned}
$$

Proof of Condition 1: This is much the same as the proof of condition 2, but our probability now looks like this (you do the math):

$$
\binom{n}{k}\binom{n/8}{5k}\left(\frac{k}{n}\right)^{10k}
$$