Lecture 5: April 17, 2013

Madhur Tulsiani Scribe: Somaye Hashemifar

1 Chernoff bounds recap

We recall the Chernoff/Hoeffding bounds we derived in the last lecture. Let Z be a sum of n independent 0/1 random variables $\{X_i\}$ and $\mathbb{E}[Z] = \mu$. Then we have

$$\mathbb{P}[Z \geq (1+\sigma)\mu] \leq \left[\frac{e^{\sigma}}{(1+\sigma)^{(1+\sigma)}}\right]^{\mu}.$$

Now, let's look at a large deviation using Chernoff bounds:

$$\begin{aligned}\mathbb{P}[Z \geq e\mu] &\leq \left(\frac{e^{e-1}}{e^{e}}\right)^{\mu} \\ &= e^{-\mu}.\end{aligned}$$

When $(1+\sigma)$ is even larger i.e. $(1+\sigma) \geq 2e$, using Chernoff bound we would have

$$\begin{aligned}\mathbb{P}[Z \geq (1+\sigma)\mu] &\leq \left(\frac{e}{1+\sigma}\right)^{(1+\sigma)\mu} \\ &\leq 2^{-(1+\sigma)\mu}.\end{aligned}$$

2 Permutation routing in hypercube

We now continue the description of the randomized routing scheme for the hypercube graph. Recall that the n-dimensional binary hypercube graph is a graph with $N = 2^n$ nodes where

- $V = \{0,1\}^n$
- $(x, y) \in E$ if $x, y \in V$ and x and y differ in just one bit.

Let $\pi : \{0,1\}^n \to \{0,1\}^n$ be any permutation of the nodes. The goal is to send a packet from node x to node $\pi(x)$, simultaneously for all nodes x. Recall that we are working in the synchronous model, where at each time instant we are allowed to send at most one packet across each edge of the hypercube. If there are multiple packets waiting to cross an edge at time t, then we put them in a queue and choose one packet to cross that edge (the method for choosing a packet out of the queue can be arbitrary). Also, we are interested in *oblivious* routing schemes, such that the path of the packet from x to $\pi(x)$ depends only on x and $\pi(x)$ and not on the destinations of any other packets. We will prove the following result due to Brebner and Valiant [BV81].

Theorem 2.1 ([BV81]) *There is a randomized oblivious scheme to route packets from each $x \in V$ to $\pi[x]$, which takes time $O(n)$ with probability $1 - 2^{-\Omega(n)}$.*

In the description below, we identify a packet with its source and refer to the packet being routed from x to $\pi(x)$ as "packet x". We now describe the routing scheme.

Scheme: The randomized routing scheme has two phases.

- **Phase 1:** Packet x chooses a random intermediate destination $\gamma(x) \in \{0,1\}^n$ and goes to $\gamma(x)$ using the "bit-fixing" path.
- **Phase 2:** Packet x goes from $\gamma(x)$to $\pi(x)$ using the "bit-fixing" path.

Recall that a "bit-fixing" path between i and j is a path obtained by flipping bits of i that are different from j in a fixed sequence to reach j. For example a bit-fixing path from $i = 0011001$ to $j = 111001$ in the hypercube $\{0,1\}^7$ is $i = 0011001 \rightarrow 1011001 \rightarrow 1111001 \rightarrow 1110001 = j$. A bit-fixing path is always a shortest path from i to j in the hypercube.

Basic idea: Since $\pi(x)$ is a permutation, the two phases are symmetric. It will be sufficient to show that the time taken in Phase 1 is $O(n)$ with high probability. Let P_x be the bit-fixing path from x to $\gamma(x)$ and $T(x)$ be the total time taken by packet x. Also, let $D(x)$ be the delay for packet going from x to $\gamma(x)$ i.e., the time spent by packet x without moving when it is waiting in a queue at some intermediate node. Then $T(x) \leq n + D(x)$.

We want to show that $D(x)$ is small for every x. The following claim provides an upper bound on $D(x)$.

Claim 2.2 *For all $x \in \{0,1\}^n$,*

$$D(x) \;\leq\; \textit{number of paths intersecting } P_x \;=\; |\{y \neq x \;:\; P_y \cap P_x \neq \emptyset\}| \;.$$

We first analyze the routing scheme assuming the above claim. For each x, we define the random variable Z_x

$$Z_x \;=\; |\{y \neq x \;:\; P_y \cap P_x \neq \emptyset\}|$$

We want to show that Z_x is $O(n)$ for each x. We first derive an upper bound on Z_x. Let P_x be the path $(e_1, \ldots, e_k)$. Then

$$Z_x \;\leq\; \sum_{i=1}^{x} \left(\#\text{ paths passing through } e_i \right) .$$

Note that in the above description the path $(e_1, \ldots, e_k)$ and its length k are also random. Since we do not want to compute any bounds conditioned on the choice of P_x, we will in fact show that with high probability, *for every bit-fixing path P*, the number of paths P_y intersecting P is $O(n)$. Fix an arbitrary bit-fixing path $P = (e_1, \ldots, e_k)$ and define

$$N_P = |\{y \in \{0,1\}^n \;:\; P \cap P_y \neq \emptyset\}| \;\leq\; \sum_i N_{e_i} \,,$$

Where an edge e_i, N_{e_i} denotes the number of paths passing through e_i. By symmetry, we must have that $\mathbb{E}[N_e]$ is the same for all $e \in E$. Then,

$$\sum_e \mathbb{E}[N_e] \;=\; \mathbb{E}\left[\sum_e N_e\right] \;=\; \mathbb{E}\left[\sum_x |P_x|\right] \;=\; 2^n \cdot \frac{n}{2}.$$

Note that in the above we are counting (x, y) and (y, x) as *different* edges. Thus, the number of edges is $2^n \cdot n$, which gives $\mathbb{E}[N_e] = 1/2$ for all e. Thus, we have for each path P,

$$\mathbb{E}[N_P] \;=\; \sum_i \mathbb{E}[N_{e_i}] \;=\; |P| \cdot \frac{1}{2} \;\leq\; \frac{n}{2}.$$

Also, N_P can be written as a sum of 2^{n-1} independent random variables $I_{P,y}$ where

$$I_{P,y} \;=\; \begin{cases} 1 & \text{if } P \cap P_y \neq \emptyset \\ 0 & \text{otherwise} \end{cases}.$$

Thus, can now bound the probability that N_P is large using Chernoff bounds. Since $2e \leq 6$, we have

$$\mathbb{P}[N_P \geq 3n] \;\leq\; 2^{-3n}.$$

Note that the number of bit-fixing paths is at most 2^{2n}, since the path can be specified by specifying a start and an end. Thus, the probability that for any P, N_P is greater than $3n$ is at most $2^{2n} \cdot 2^{-3n} = 2^{-n}$. Also, for any x, we have that $Z_x \leq N_{P_x} \leq 3n$. Thus, with probability at least $1 - 2^{-n}$, the time taken by all packets in phase 1 is at most $n + 3n = 4n$. We now prove Claim 2.2

Proof of Claim 2.2: First note that, in the hypercube, when two bit-fixing paths diverge they will not come together again; i.e. paths which intersect will intersect only in one contiguous segment. Let $P_x = (e_1, , \ldots, e_k)$. For every packet y which goes through P_x (at least for some time), define the *lag* of packet y at the start the time step t is as $t - i$, where e_i is the next edge that packet y wants to traverse. Note that at time t, all packets waiting to cross an edge must have the same lag. At each time step, one of these packets crosses the edge and its lag remains unchanged (or it exits the path) and the lag of the remaining packets increases by 1.

The lag of the packet x goes from 0 to $D(x)$. We will charge each increase in the lag of packet x to a *unique* y such that $P_y \cap P_x \neq \emptyset$. Consider the time step t when the lag of packet x goes from L to $L + 1$; suppose this happens when it is waiting to traverse edge e_i. Then x must be held up in the queue at e_i (else its lag would not increase), so there exists at least one other packet at e_i with lag L, and this packet actually moves at step t. For any such packet, at each step either it moves and its lag remains L, or it waits an edge and some other packet with lag L crosses the edge. Now consider the last time at which there exists a packet y with lag L. This packet must exit the path or its lag will remain L. We can charge this packet y for increasing the lag of packet x from L to $L + 1$. Each packet y is charged at most once, because it is charged only when it leaves P_x which by the observation at the start of the proof, happens only once. ■

3 Balanced Allocations

We consider the following problem of allocating jobs to servers: We are given a set of n servers $1, \ldots, n$ and m jobs arrive one by one. We seek to assign each job to one of the servers so that the loads placed on all servers are as balanced as possible.

In developing simple, effective load balancing algorithms, randomization often proves to be a useful tool. We consider two approaches for this problem:

- **Random Choice**: one possible way to distribute the jobs is to simply place each job on a random server, chosen independently of the previous allocations.
- **Two Random Choices**: For each job, we choose two servers independently and uniformly at random and place the job on the server with less load (breaking ties arbitrarily).

We will show that using two random choices significantly reduces the maximum load on any server. For the entire analysis, we will work with the case when $m = n$. The analysis easily extends to an arbitrary m, but it easier to appreciate the bounds when $m = O(n)$ (and in particular when $m = n$).

It is convenient to think of the above in terms of the so called "balls and bins" model. Each job can be thought of a s ball and each server is a bin. We think of assigning job j to a server i as throwing the j^{th} ball in bin i. The load of a server is the same as the number of balls in the corresponding bin.

3.1 Random choice

Suppose $Z_i =$ number of balls in bin i. We can write

$$Z_i = \sum_j X_{ij}, \quad \text{where} \quad X_{ij} = \begin{cases} 1 & \text{if ball } j \text{ is thrown in bin } i \\ 0 & \text{otherwise} \end{cases} .$$

Then, we have that each Z_i is a sum of $m(= n)$ independent random variables with $\mathbb{E}[Z_i] = 1$. Let $K = \frac{\ln n}{\ln \ln n}$. By Chernoff/Hoeffding bounds, we have that for each i,

$$\mathbb{P}[Z_i \geq K] \;\leq\; \left(\frac{e}{K}\right)^K .$$

Thus, the probability that there exists an i such that $Z_i \geq K$ is at most $n \cdot \left(\frac{e}{K}\right)^K$, which is at most $\frac{\ln n}{n}$ for the above value of K. Hence, with probability at least $1 - \frac{\ln n}{n}$, the maximum number of balls in a bin is at most $\frac{\ln n}{\ln \ln n}$.

3.2 The power of two random choices

We will now show that two random choices can reduce the maximum load to $O(\ln \ln n)$. The proof technique is due to Azar et al. [ABKU94] and various applications were explored by Mitzenmacher in his thesis [Mitz96]. We first provide the intuition for the proof.

For each i, let B_i denote the number of bins with at least i balls. Suppose $B_i \leq \beta_i$ for some bound β_i. Then B_{i+1} is bounded above by a binomial random variable corresponding to the number of heads in n independent coin tosses, where the probability of each toss being heads is at most $(\beta_i/n)^2$. This is because for a ball to land a bin such that the load of the bin becomes greater than i, it must happen that both the random bins which we chose to put it in, had load at least i. This happens with probability at most $(\beta_i/n)^2$. Thus, B_{i+1} is upper bounded by the above random variable, which we denote as $\mathsf{Bin}\left(n, \left(\frac{\beta_i}{n}\right)^2\right)$.

This, $\mathbb{E}[B_{i+1}] \leq n \cdot \left(\frac{\beta_i}{n}\right)^2$ and B_{i+1} is at most $e \cdot \frac{\beta_i^2}{n}$ with high probability. We can then take β_{i+1} to be $e \cdot \frac{\beta_i^2}{n}$. For the above sequence, the value of β_i becomes less than 1 for $i_0 = O(\ln\ln n)$, and thus we can bound the maximum load by i_0. The proof will follow this intuition, except that for the last step, when $\mathbb{E}[B_i]$ becomes very small, we will not be able to use a Chernoff bound and will have to resort to a slightly different analysis.

We first define the values β_i. Let $\beta_6 = \frac{n}{2e}$ and $\beta_{i+1} = e \cdot n \cdot \left(\frac{\beta_i}{n}\right)^2$.

$$
\begin{aligned}
& \beta_6 = \frac{n}{2e} \\
\Rightarrow \quad & \beta_7 = e\left(\frac{n}{2e}\right)^2 n = \frac{n}{4e} = \frac{n}{2^2 e} \\
\Rightarrow \quad & \beta_8 = e\left(\frac{n}{4e}\right)^2 n = \frac{n}{16e} = \frac{n}{2^{2^2} e} \\
\Rightarrow \quad & \beta_9 = e\left(\frac{n}{16e}\right)^2 n = \frac{n}{256e} = \frac{n}{2^{2^3} e} \\
& \vdots \\
\Rightarrow \quad & \beta_i = \frac{n}{2^{2^{i-6}} e}
\end{aligned}
$$

Let E_i be the event that $B_i \leq \beta_i$. Note that E_6 holds for sure since there can be at most $n/6 \leq n/2e$ bins with 6 or more balls. We show that with high probability, if E_i holds then E_{i+1} holds provided $\beta_i^2 \geq 2n \ln n$.

Claim 3.1 *Let i be such that $\beta_i^2 \geq 2n\ln n$. Then,*

$$
\mathbb{P}[\neg E_{i+1} \mid E_i] \;\leq\; \frac{1}{n^2} \cdot \frac{1}{\mathbb{P}[E_i]}.
$$

Proof: The proof follows from a direct calculation. We have

$$
\begin{aligned}
\mathbb{P}[\neg E_{i+1} \mid E_i] &= \frac{\mathbb{P}[\sim E_{i+1} \wedge E_i]}{\mathbb{P}[E_i]} \\
&\leq \frac{\mathbb{P}\left[\mathsf{Bin}\left(n, \left(\frac{\beta_i}{n}\right)^2\right) \geq en\left(\frac{\beta_i}{n}\right)^2\right]}{\mathbb{P}[E_i]} \\
&\leq \frac{e^{-n\cdot(\beta_i/n)^2}}{\mathbb{P}[E_i]} \;\leq\; \frac{1}{n^2} \cdot \frac{1}{\mathbb{P}[E_i]}
\end{aligned}
$$

when $\beta^2 \geq 2n \ln n$. ■

We can then use induction to show that for each i as above, the probability of the event E_i not happening is very low.

Claim 3.2 *For all i such that $\beta_i^2 \geq 2n \ln n$, we have*

$$\mathbb{P}\left[\neg E_{i+1}\right] \;\leq\; \frac{i+1}{n^2}.$$

Proof: We prove the claim by induction on i. We know from the definition of β_6 that $\mathbb{P}\left[\neg E_6\right] = 0$. Also, from the previous claim, we have that for any i as above,

$$\begin{aligned}
\mathbb{P}\left[\neg E_{i+1}\right] &= \mathbb{P}\left[E_i\right] \cdot \mathbb{P}\left[\neg E_{i+1} | E_i\right] \;+\; \mathbb{P}\left[\neg E_i\right] \cdot \mathbb{P}\left[\neg E_{i+1} | \neg E_i\right] \\
&\leq \mathbb{P}\left[E_i\right] \cdot \frac{1}{n^2} \cdot \frac{1}{\mathbb{P}\left[E_i\right]} \;+\; \frac{i}{n^2} \\
&\leq \frac{i+1}{n^2}.
\end{aligned}$$

■

We will need a slightly different analysis when $\beta_i^2 < 2n \ln n$, which we will cover in the next lecture.

References

[BV81] G. Brebner and L. Valiant, "Universal Schemes for Parallel Communication", *Proceedings of the 13th Annual ACM Symposium on Theory of Computing*, 1981, pp. 263–277.

[ABKU94] Y. Azar, A. Z. Broder, A. R. Karlin and E. Upfal, "Balanced Allocations", *Proceedings of the 26th Annual ACM Symposium on Theory of Computing*, 1994, pp. 593–602.

[Mitz96] M. Mitzenmacher, "The Power of Two Choices in Randomized Load Balancing", PhD thesis, University of California Berkeley, 1996.