Kleinberg-Tardos Section 13.10 Load balancing

Goal assign m jobs to n processors so that there as loaded as evenly as possible

Must assign a job to a processor when it arrives.

If we had full control, then we could achieve a maximum load of $\left\lceil \frac{m}{n} \right\rceil$

Idea: assign a new job to a random processor

$\Downarrow$ Expected # jobs at any given processor is $\frac{m}{n}$

(see next page)

Question: How close can we get to this value?

Analysis depends on the relative sizes of m and n.

Case 1 $m = n$

Expect one job per processor:

X_i = # jobs given to processor i $\quad i = 1, 2, \ldots, n$

$$Y_{ij} = \begin{cases} 1 & \text{if job } j \text{ assigned to processor } i \\ 0 & \text{else} \end{cases}$$

$$E(Y_{ij}) = p(Y_{ij} = 1) = \frac{1}{n}$$

$$X_i = \sum_{j=1}^{m} Y_{ij} \quad \text{so}$$

$$E(X_i) = E\left(\sum_{j=1}^{m} Y_{ij}\right) = \sum_{j=1}^{m} E(Y_{ij}) = \sum_{j=1}^{m} \frac{1}{n} = \frac{m}{n} = 1$$

Recall Chernoff bounds

$$\Pr(X > (1+\delta)\mu) \leq \left[\frac{e^{\delta}}{(1+\delta)^{(1+\delta)}}\right]^{\mu}$$

$$(\square) \quad \Pr(X_i > c) < \left[\frac{e^{c-1}}{c^c}\right]^1$$

here $\mu = 1$ and $(1+\delta) = c$

We want $P_r(X_i > c)$ to be at so small that we can apply the Union bound and still get a low probability that there is <u>any</u> i s.t. $X_i > c$

In particular we want to choose c such that $P_r(X_i > c) << \frac{1}{n}$

study the function c^c

$$\boxed{\text{Question what is the solution to } x^x = n\text{?}}$$

Call this number $\gamma(n)$ (so $\gamma(n)^{\gamma(n)} = n$

$$x^x = n \Rightarrow x \log x = \log n \Rightarrow \log x + \log\log x = \log\log n$$

This implies that $2\log x > \log\log n > \log x$
and using that $x \log x = \log n$ we get

$$\boxed{\frac{1}{2}x < \frac{\log n}{\log\log n} < x = \gamma(n)}$$

$$\boxed{\frac{1}{2}x < \frac{\log n}{\log\log n} < x = \gamma(n)}$$

$$\Downarrow \quad \gamma(n) = \Theta\left(\frac{\log n}{\log\log n}\right)$$

Take $c = e\gamma(n)$ in (B):

$$\Pr(X_i > c) < \frac{e^{c-1}}{c^c} < \left(\frac{e}{c}\right)^c = \left(\frac{1}{\gamma(n)}\right)^{e\gamma(n)} < \left(\frac{1}{\gamma(n)}\right)^{2\gamma(n)} = \left(\frac{1}{\gamma(n)^{\gamma(n)}}\right)^2 = \frac{1}{n^2}$$

By the union bound the probability that some X_j is larger than $c = e\gamma(n)$ is less than $n \cdot \frac{1}{n^2} = \frac{1}{n}$

So with probability at least $1 - \frac{1}{n}$ all processors receive at most $e \cdot \gamma(n) = \Theta\left(\frac{\log n}{\log\log n}\right)$ jobs

It can be shown (not proven) that with high probability some processor will receive $\Theta\left(\frac{\log n}{\log\log n}\right)$ jobs so the bound is asymptotically tight

How fast does $f = \frac{\log n}{\log\log n}$ grow?

SLOW! with $n = 2^{2^{10}} = 2^{1024}$ we have $f = \frac{1024}{10} \sim 102$

Case 2 $m > n$

When m increases compared to n, the load smoothens out rapidly:

Suppose $m = 16 n \ln n$

Then $E(X_i) = 16 \ln n$ and we have

$$\Pr(X_i > 2\mu) \le \left(\frac{e^1}{2^2}\right)^\mu$$

$$= \left(\frac{e}{4}\right)^{16 \ln n}$$

$$\left(\frac{e}{4}\right)^8 < \frac{1}{e}$$

$$\le \left(\frac{1}{e}\right)^{\ln n}$$

$$= \frac{1}{n^2}$$

$$\Pr\left(X_i < \tfrac{1}{2}\mu\right) \le e^{-\frac{1}{2}\left(\frac{1}{2}\right)^2 \cdot (16 \ln n)} = e^{-2\ln n} = \frac{1}{n^2}$$

$$\boxed{\text{So when there are } n \text{ processors and } \Omega(n \log n) \text{ jobs with high probability all processors have load between half and twice the average load.}}$$