0.1 Introduction

Here are some of the typical uses of randomness in parallel computation.

The first use is for coordinating processors; usually random methods are faster than deterministic ones, or in someways simpler. For example, if two processors contend for the same resource, then we can flip a coin to decide who gets it.

A more general application is what is called "symmetry breaking" – there are some P processors wanting one of some k resources. The problem is how to coordinate who gets what, and so that as many processors as possible are given a resource. One idea would be to each processor make a request to a random resource: this would distribute the processors reasonably evenly amongst the resources, and then use coin tossing to remove further contention. In this case, deterministic methods might require more computation, or might not distribute the load uniformly.

A related use is in packet routing algorithms, where we use random priorities for messages. This effectively puts messages into batches, with each batch getting roughly the same number on the average. Notice that this is a strong statement: by randomly assigning messages into batches we are likely to get a good partitioning on all links in the system. Whereas it is very easy for deterministic methods to get a perfectly good partitioning on a single link, but doing it simultaneously on all links is quite hard.

1 Probability

Probability space is defined as the set of all the possible outcomes of an experiment. This is also known as *sample space*. Let us denote the sample space by $\mathcal{P}$. Any subset of the probability space is known as an *event*.

Example 1: If we perform the experiment of flipping two coins then the probability space is $C_2 = \{tt, th, ht, hh\}$. The event that there is atleast one tail is given by $\{tt, th, ht\}$.

Example 2: If the experiment performed is drawing a card from the deck, then the probability space will be the entire deck (D).

A *probability distribution* is a function P from the set of events to $\Re$. This function should satisfy the following axioms of probability.

Axioms of Probability

1. For any event A, $P(A) \geq 0$
2. $P(\mathcal{P}) = 1$,
3. For any infinite sequence of disjoint events $A_1, A_2, \ldots$,

$$P\left(\bigcup_{i=1}^{\infty} A_i\right) = \sum_{i=1}^{\infty} P(A_i)$$

1.1 Basic Properties of Probability

1. $P(\emptyset) = 0$.

2. For any finite sequence of disjoint events $A_1, A_2, \ldots A_n$

$$P\left(\bigcup_{i=1}^{n} A_i\right) = \sum_{i=1}^{n} P(A_i)$$

3. For any event A, $P(A^c) = 1 - P(A)$.

4. For any event A, $0 \leq P(A) \leq 1$.

5. If $A \subset B$, then $P(A) \leq P(B)$.

1.2 Random Variables

A *random variable* X is a function from the probability space to the set of real numbers. Formally, $X : \mathcal{P} \to \Re$.

Example 1: Let H_2 be the no. of heads in 2 coin flips, hence in our sample space C_2, we have, $H_2(tt) = 0, H_2(ht) = 1, H_2(th) = 1, H_2(hh) = 2$.

Example 2: Let C be 1 if the card drawn is a club, 0 otherwise. Let A be 1 if the card drawn is an ace, 0 otherwise.

It is common to define the event $X = r$ to be the set $\{x \in \mathcal{P} | X(x) = r\}$

Example 1: Define a random variable F to be 1 if the first toss is a head, and 0 otherwise, in our probability space C_2. And similarly define S to be 1 if the second toss is a head and 0 otherwise. Clearly $F = 0 \equiv \{tt, th\}$ and so on. A random variable X is said to be *discrete* if it takes at most countable values $x_1, x_2, \ldots$ in $\Re$. For such random variables we can define a *probability function* $f(x)$ such that,

$$f(x) = P(X = x)$$

and hence $\sum_{i=1}^{\infty} f(x_i) = 1$. Both H_2 and C are discrete random variables. Henceforth, all random variables in these notes are discrete unless specified otherwise.

A random variable that takes values only 0 and 1 is said to be a *Bernoulli random variable.* Bernoulli random variables are encountered very commonly and as will be seen soon, can often be seen as building blocks for more complex random variables.

2 Probabilistic analysis

Typically, in probabilistic analysis, we express the quantity as a random variable, and try to get estimates about its value. In this course, most commonly we will be arguing that a certain random variable will not be too large. There are various ways of going about this: (i) Estimate the probability that the variable takes large values, (ii) Estimate the value that the variable takes on the *average*, where the average is taken over the probability space. Technically this is called the expectation of the random variable.

2.1 Expectation of a Random Variable

The *expectation* of a random variable X, denoted by $E[X]$ is defined as,

$$E[X] = \sum_{x} xP(X = x)$$

$E[X]$ is also known as the *expected value* of X or the *mean* of X.

Example : We will calculate the expected value of H_2 assuming uniform probability distribution over C_2, i.e tt, th, ht, and hh all have probability $1\frac{}{4.}to$
E[H$_2$] $= 0 \cdot P(H_2 = 0) + 1 \cdot P(H_2 = 1) + 2 \cdot P(H_2 = 2)$
$= 0 + \frac{1}{2} + \frac{1}{2}$
$= 1$
$E[F] = 0 \cdot P(F = 0) + 1 \cdot P(F = 1)$
$= P(F = 1)$
$= \frac{1}{2}$

We note that the statement $E[F] = P(F = 1)$ holds for all Bernoulli random variables: the expectation of any Bernoulli random variable is the same as the probability that it takes value 1.

It is sometimes easier to estimate the expectation of a random variable if it can be thought of as a *sum of random variables.*

2.2 Sums of Random Variables

Suppose X, Y and Z are random variables satisfying $Z(x) = X(x) + Y(x)$ for all $x \in \mathcal{P}$. Then Z is said to be the sum of random variables X and Y and we write $Z = X + Y$. Expectation distributes over addition.

Lemma 2.1 (Linearity of Expectation) *If $Z = X + Y$ then*

$$E[Z] = E[X] + E[Y]$$

The proof is an easy exercise.

Example : It is easily seen that $H_2 = F + S$. Now from linearity of expectation it follows that

$$E[H_2] = E[F] + F[S] = \frac{1}{2} + \frac{1}{2} = 1$$

2.3 Markov's inequality

Theorem 2.2 (Markov's inequality) *If a random variable X only takes non-negative values, then*

$$P(X \geq k) \leq \frac{E[X]}{k}$$

The proof is an easy exercise.

Example: Suppose we toss 100 fair coins. Then clearly the expected number of heads is 50. Markov's inequality says that the probability of getting 75 heads is at most $\frac{50}{75} = \frac{2}{3}$.

Markov's inequality usually gives us a fairly weak bound. We can usually get tighter results if the random variable of interest is made up of sums of *independent* random variables.

2.4 Independent Random Variables

Two random variables X and Y are said to be *independent* if for any two real numbers x and y,

$$P(X = x \text{ and } Y = y) = P(X = x) \cdot P(Y = y)$$

In other words, two random variables are independent if knowing the value of one doesn't help us predict the value of the other variable.

Example: F and S are independent. So are C and A (Section 1.2).

2.5 Chernoff Bounds

Suppose a random variable $X = X_1 + X_2 + \cdots + X_n$, where X_i are *independent Bernoulli random variables* with $P(X_i = 1) = p_i$.

Let $\mu = E[X] = p_1 + p_2 \cdots + p_n$

Then,

$$\begin{array}{rcll} P(X \geq \beta\mu) & \leq & e^{(1-\frac{1}{\beta}-\ln\beta)\beta\mu} & \beta \geq 0 \\ & \leq & \left(\dfrac{\beta}{e}\right)^{-\beta\mu} & \beta \geq 0 \\ P(X \geq (1+\epsilon)\mu) & \leq & e^{-\epsilon^2\mu/3} & 0 < \epsilon < 1 \\ P(X \geq (1+\epsilon)\mu) & \leq & e^{-\epsilon^2\mu/4} & 0 < \epsilon < 2e-1 \\ P(X \geq (1+\epsilon)\mu) & \leq & 2^{-(1+\epsilon)\mu} & 2e-1 \leq \epsilon \\ P(X \leq (1-\epsilon)\mu) & \leq & e^{-\epsilon^2\mu/2} & 0 < \epsilon \end{array}$$

Alternatively, let $m = \beta\mu$,

$$P(X \geq m) \leq \left(\frac{m}{\mu e}\right)^{-m} \tag{1}$$

2.6 Proof idea

We only prove equation (1), and that too under the assumption that the X_i have identical distributions. That is, let $p = p_1 = p_2 = \cdots = p_n$. Hence, $\mu = np$.

The value taken by the random variable X will be greater than m, if atleast m variables out of the n take the value 1. We will estimate the probability of this event and show that it is less than the right hand side of (1).

To make atleast m variables take the value 1, we choose m variables from n and make them take the value 1. The probability of this event is $\binom{n}{m}p^m$. This will result in a lot of double counting, as we are allowing the other variables to take any value. But this will still give us a an upper bound on the probability.

We use the standard inequality,

$$\binom{n}{m} \leq \left(\frac{ne}{m}\right)^m \tag{2}$$

Hence we get

$$\begin{aligned} P(X \geq m) &\leq \binom{n}{m} p^m \\ &\leq \left(\frac{npe}{m}\right)^m \text{ using (2)} \\ &= \left(\frac{\mu e}{m}\right)^m \end{aligned}$$

□

Thus we have proved (1) for this special case of all X_i being identical.

3 With High Probability

In this section we will define the notion of an upper bound on a random variable *with high probability.* Let N be the size of the problem. Let $f(N)$ be the random variable under consideration. Let g be a function of one variable. We say that,

$$f(N) = O(g(N)) \text{ with high probability (w.h.p)}$$

if and only if, there exist a constant N_0 and a function c such that

$$P(f(N) > c(k)g(N)) \leq N^{-k} \tag{3}$$

for any k whenever $N \geq N_0$.

3.1 Compositional Properties

Suppose A_i, $i = 1, \ldots, m$ are random variables, arising in a certain problem, with N denoting the problem size. Suppose further that m is atmost polynomially large in N, i.e. $m \leq N^a$ Suppose we know that $A_i = O(g(N)$ w.h.p. for $i = 1 \ldots m$. Define

$$\begin{aligned} A_{series}(N) &= \sum_i A_i(N) \\ A_{parallel}(N) &= \max_i A_i(N) \end{aligned}$$

Then,

$$\begin{aligned} A_{series}(N) &= O(mg(N)) \text{ w.h.p.} \\ A_{parallel}(N) &= O(g(N)) \text{ w.h.p.} \end{aligned}$$

The proofs are as follows. We know that there exist c_i, N_0 such that $\Pr[A_i > c_i(k)g(N)] \leq N^{-k}$. Define $c = \max_i c_i$. Then we have $\Pr[A_i > c(k)g(N)] \leq N^{-k}$.

$$\Pr[A_{series}(N) > mc(k)g(N)] = \Pr[\sum_i A_i > mc(k)g(N)] \leq \Pr[\text{Some } A_i > c(k)g(N)] \leq mN^{-k} = N^{-(k-a)}$$

Setting $k' = k - a$ and $c'(k') = c(k' + a) = c(k)$ we see that

$$\Pr[A_{series}(N) > mc'(k')g(N)] \leq N^{-k'}$$

Thus $A_{series}(N) = O(mg(N))$ w.h.p.

4 Exercises

1. A commonly encountered experiment is that of throwing B balls into N buckets. Each ball is thrown into a random bucket chosen uniformly and independently. The main question is the number of balls in the most filled bucket. It is instructive to consider 3 cases.

   (a) $B = N$. In this case show that the number of balls in the most filled bucket is $O(\log N/\log\log N)$ w.h.p. It will be useful to note that $\log N/\log\log N \geq \sqrt{\log N}$ during the algebraic manipulation.

   (b) $B = N\log N$. Show that the size of the most filled will be $O(\log N)$ w.h.p.

   (c) $B = N^2$. This time try to get the constants as small as possible too.

   The point of the above exercises is to observe that as the number of balls increases w.r.t. the number of buckets, the distribution gets closer to the perfect distribution.

2. Suppose we have a $n \times n$ mesh with $N = n^2$. Suppose each processor has p keys. Suppose further that each key is a randomly chosen real number from [0,1). You are to analyze the time taken for the following bucket sort like algorithm. The final order is not snakelike, but row major.

   (a) Processor (i,j) sends all its keys in the range $[\frac{k}{n}, \frac{k+1}{n})$ to processor (k, j) for all k $(0 \leq k < n)$.

   (b) Processor (k, j) now only has keys in $[\frac{k}{n}, \frac{k+1}{n})$. From these it sends the keys in range $[\frac{k}{n} + \frac{i}{n^2}, \frac{k}{n} + \frac{i+1}{n^2})$ to processor (k, i).

   Show that the first step can be done in time $O(pn)$ deterministically. Show that the second can be done in time $O(pn)$ with high probability.

3. Suppose you are given a circuit consisting of gates with at most d inputs and one output. Each gate works as follows: it waits until all inputs are available and then generates the output (which is connected to the input of at most one gate). The gates in this problem behave probabilistically (this is to model asynchronous circuit behaviour). The time required to generate the output after all the inputs are available is an independent random variable, taking the value $1 + \epsilon$ with probability p and the value ϵ with probability $1 - p$. The parameters ϵ and p are common to all gates, but may depend upon, say the number of gates. In other words, in the analysis they should not be treated as constants. The delay of a circuit is defined as the time taken by the output to be generated after all inputs have arrived.

   (a) Consider a circuit which is just a long chain of h unary gates. There is a single input and single output. In the best case, the output will have a delay of $h\epsilon$, and in the worst case, the output will have a delay of $h + h\epsilon$. What is the expected delay of the output? (b) What is the probability that the delay in the above circuit will be at least $h/2$ (c) Consider a circuit whose graph is a directed tree, with at most h gates are present on any path from any input to the output. The gates may have degree upto d, but the number of inputs is N. Show that the circuit delay is $O(h\epsilon + h\epsilon + \log N)$ with high probability. Give a delay sequence argument.

4. Suppose each processor (i, j, k) in an $n \times n \times n$ mesh sends a packet to $\rho(i, j, k)$, where $\rho(i, j, k)$ is chosen to be some unique processor within a distance d in

the mesh. Suppose the packet is sent by correcting the coordinates in left to right order.

(a) Will this cause deadlocks if standard buffer management is used? (b) What is the maximum congestion possible? (c) If $\rho(i, j, k)$ is chosen independently at random from all possible processors within a distance d of (i, j, k), what is the expected congestion in each link? What can you say with high probability?