CS 4810/7800: Advanced Algorithms	Spring 2026
Lecture 9 – February 9, 2026	
Prof. Prashant Pandey	*Scribe: Avni Mishra*

1 Overview

In this lecture, we review basic probability concepts and introduce the balls and bins problem, which is fundamental to understanding hashing and load balancing.

2 Motivation: Load Balancing

2.1 The Problem

When you type in `google.com`, your request goes into a web server. To avoid the following issues, there are multiple web servers that replicate the same data:

- *Fail-safe for servers* – ensure availability when servers go down
- *Load balancing* – distribute queries evenly so no single server becomes a bottleneck
- *Geographic dispatches* – route to nearby servers to minimize latency

However, this raises the question: how do we distribute incoming requests among these servers?

2.2 Simple Solutions

The naive approach would be to query all server loads and route each request to the least loaded server. However, maintaining a data structure that tracks server loads requires constant updates and fast minimum queries. This would most likely mean maintaing a min heap, for which insertions are a cost of $O(\lg n)$. This creates a logarithmic bottleneck that defeats the purpose of load balancing.

3 Review of Probability Theory

Definition 1. *A probability sample space* $(S, \mathbb{P})$ *consists of:*

- $S = \{s_1, s_2, s_3, \ldots\}$ *is the set of all possible outcomes*
- $\mathbb{P} : S \to [0, 1]$ *is a probability function where* $\sum_i \mathbb{P}(s_i) = 1$

Definition 2. *An event is a subset of outcomes from a sample space* $(S, \mathbb{P})$*.*

Definition 3. *A random variable is a function defined as:*

$$f : S \to \mathbb{R}^k$$

which takes the set of outcomes and maps it to a number (or vector).

Definition 4. *The expected value $E[f]$ of a random variable f is:*

$$E[f] = \sum_{s_i \in S} \mathbb{P}(s_i) \cdot f(s_i)$$

Definition 5 (Linearity of Expectation)**.** *For any random variables f and g:*

$$E[f + g] = E[f] + E[g]$$

Definition 6. *The conditional probability $\mathbb{P}(A \mid B)$ denotes the probability of event A happening given that event B is happening:*

$$\mathbb{P}(A \mid B) = \frac{\mathbb{P}(A \cap B)}{\mathbb{P}(B)}$$

Definition 7. *Events A and B are independent if and only if:*

$$\mathbb{P}(A \cap B) = \mathbb{P}(A) \cdot \mathbb{P}(B)$$

When A and B are independent:

$$\mathbb{P}(A \mid B) = \frac{\mathbb{P}(A) \cdot \mathbb{P}(B)}{\mathbb{P}(B)} = \mathbb{P}(A)$$

Definition 8. *Events $E_1, \ldots, E_n$ are mutually independent if and only if for every subset of the events, the probability of their intersection equals the product of their individual probabilities:*

$$\mathbb{P}(E_1 \cap E_2 \cap \cdots \cap E_n) = \mathbb{P}(E_1) \cdot \mathbb{P}(E_2) \cdots \mathbb{P}(E_n)$$

Definition 9. *Events $E_1, \ldots, E_n$ are pairwise independent if and only if for every pair of distinct events:*

$$\mathbb{P}(E_i \cap E_j) = \mathbb{P}(E_i) \cdot \mathbb{P}(E_j) \quad \text{for all distinct } i, j$$

Observation 10. *If events are mutually independent, then they are also pairwise independent. However, the converse is not true: pairwise independence does not necessarily imply mutual independence. In practice, most hashing schemes rely on pairwise independence, which is often sufficient for algorithmic purposes.*

Definition 11. *Let E_n be an event on problem size n. We say E_n occurs With High Probability (WHP) if:*

$$\mathbb{P}[E_n] \geq 1 - \frac{1}{n^c}$$

for some constant $c > 1$.

As n becomes large, the probability that E_n occurs approaches 1, meaning the event is essentially guaranteed to happen.

3.1 Disjoint Events

Question: Let A and B be two disjoint events. Are these independent?

Answer: *NOT independent.* Consider:

$$\mathbb{P}(A \cap B) = 0$$
$$\mathbb{P}(A) \cdot \mathbb{P}(B) > 0$$

Thus $\mathbb{P}(A \cap B) \neq \mathbb{P}(A) \cdot \mathbb{P}(B)$. If both $\mathbb{P}(A)$ and $\mathbb{P}(B) \to 0$, this becomes pathological.

Remark. Probability is beautiful but unituitive. The following approach helps solve most probability problems:

1. Find the sample space
2. Define events of interest
3. Determine outcome probabilities
4. Determine event probability

4 Introduction to Balls-and-Bins

4.1 Problem Definition

In the balls-and-bins problem, we throw n balls equiprobably and independently into n bins (often $b = n$ in this course).

4.2 Applications

1. *Hashing* – We can gain insight into hashing by studying balls in bins, since hashing is modeled by randomly throwing data into hash table blocks.
2. *Load Balancing* – In the context of distributed systems, bins represent servers serving requests and balls represent client requests.

4.3 Key Questions

Given n balls thrown into n bins, what are the:

1. Expected number of balls in a given bin?
2. Expected number of balls in the fullest bin?

3. Expected number of balls needed to achieve at least 1 collision?
4. Expected number of empty bins?
5. Expected number of bins with a collision?
6. Expected number of balls needed to fill all bins?

In the latter parts of this lecture, we will modify these questions by replacing "Expected" with "With High Probability (WHP)."

5 Analysis with Coin Toss

5.1 Question: How many flips to expect a certain result?

Setup. Given a fair coin, how many times must we flip the coin in order to expect at least one Head result?

Solution. Define a random variable X_i for each coin i:

$$X_i = \begin{cases} 1 & \text{if heads} \\ 0 & \text{otherwise} \end{cases}$$

The probability that any coin flip equals heads is $\frac{1}{2}$. Thus:

$$E[X_i] = \sum_{s_i \in S} \mathbb{P}(s_i) \cdot f(s_i)$$

$$E[X_i] = 1 \cdot \frac{1}{2} + 0 \cdot \left(1 - \frac{1}{2}\right) = \frac{1}{2}$$

In order we determine how many flips it will take to have our Expected value $= 1$, we must determine when $E[X_0 + X_1 + X_2 \cdots + X_n] = 1$.

From the Linearity of Expectation:

$$E[\sum X_n] = \sum E[X_n]$$

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

Thus, we must flip the coin twice in order to expect a value of heads.

6 Analysis of Balls-and-Bins

6.1 Question 1: Expected Number of Balls in a Single Bin

Setup. We have n balls thrown independently and uniformly at random into n bins. What is the expected number of balls in bin 1?

Solution. Define a random variable X_i for each ball i:

$$X_i = \begin{cases} 1 & \text{if ball } i \text{ goes to bin 1} \\ 0 & \text{otherwise} \end{cases}$$

The probability that any single ball lands in bin 1 is $\frac{1}{n}$. Thus:

$$E[X_i] = \sum_{s_i \in S} \mathbb{P}(s_i) \cdot f(s_i)$$

$$E[X_i] = 1 \cdot \frac{1}{n} + 0 \cdot \left(1 - \frac{1}{n}\right) = \frac{1}{n}$$

Let $X = \sum_{i=1}^{n} X_i$ denote the total number of balls in bin 1. By linearity of expectation:

$$E[X] = \sum_{i=1}^{n} E[X_i] = n \cdot E[X_i] = n \cdot \frac{1}{n} = 1$$

Conclusion. The expected number of balls in any given bin is exactly $\boxed{1}$.

6.2 Question 2: Expected Number of Balls in the Fullest Bin (WHP)

Approach. First, we compute the probability that a single bin has at least ℓ balls, then union over all n bins.

Claim 12. *The probability that bin 1 contains ℓ balls is:*

$$\mathbb{P}[\textit{bin 1 has } \ell \textit{ balls}] = \binom{n}{\ell} \left(\frac{1}{n}\right)^{\ell} \left(1 - \frac{1}{n}\right)^{n-\ell}$$

Claim 13. *The probability that bin 1 contains at least ℓ balls is:*

$$\mathbb{P}[\textit{bin 1 has} \geq \ell \textit{ balls}] \leq \binom{n}{\ell} \left(\frac{1}{n}\right)^{\ell}$$

Given that for any number x and y,

$$\left(\frac{y}{x}\right)^{x} \leq \binom{y}{x} \leq \left(\frac{ey}{x}\right)^{x}$$

We ascertain:

$$\binom{n}{\ell} \leq \left(\frac{en}{\ell}\right)^{\ell}$$

Therefore:

$$\mathbb{P}[\text{bin 1 has} \geq \ell \text{ balls}] \leq \binom{n}{\ell}\left(\frac{1}{n}\right)^{\ell} \leq \left(\frac{en}{\ell n}\right)^{\ell} = \left(\frac{e}{\ell}\right)^{\ell}$$

$$\mathbb{P}[\text{bin 1 has} \geq \ell \text{ balls}] \leq \left(\frac{e}{\ell}\right)^{\ell}$$

Union Bound. By the union bound, the probability that *any* bin has at least ℓ balls is at most:

$$\mathbb{P}[\text{any bin has} \geq \ell \text{ balls}] \leq n\left(\frac{e}{\ell}\right)^{\ell}$$

First, let's show that $c = \lg n$ is a WHP bound on the number of balls in a bin.

Theorem 14. *With high probability, the fullest bin contains at most $O(\frac{\lg n}{\lg \lg n})$ balls.*

Proof. Setting $\ell = c \lg n$ for a large constant c:

$$\mathbb{P}[\text{any bin has} \geq c \lg n \text{ balls}] \leq n\left(\frac{e}{c \lg n}\right)^{c \lg n}$$

For sufficiently large c, $\frac{e}{c \lg n} < \frac{1}{2}$, giving:

$$\mathbb{P}[\text{any bin has} \geq c \lg n \text{ balls}] \leq n \cdot \left(\frac{1}{2}\right)^{c \lg n} = n \cdot 2^{-c \lg n} = n \cdot n^{-c} = n^{1-c}$$

Since $c > 1$, we have $n^{1-c} < \frac{1}{n^{c-1}}$, which goes to 0 as n grows. Thus the fullest bin has $O(\lg n)$ balls WHP when $\ell = c \lg n$.

$\square$

Now setting $\ell = \frac{c \lg n}{\lg \lg n}$ for a large constant c gives:

$$\mathbb{P}[\text{any bin has} \geq \frac{c \lg n}{\lg \lg n} \text{ balls}] \leq n\left(\frac{e}{\frac{c \lg n}{\lg \lg n}}\right)^{\frac{c \lg n}{\lg \lg n}}$$

$$\leq n\left(e \cdot \frac{\lg \lg n}{c \lg n}\right)^{\frac{c \lg n}{\lg \lg n}}$$

$$\leq n\left(\frac{1}{2}\right)^{c \lg n - o(c \lg n)}$$

$$\leq n^{2-c}$$

Since $c > 2$, we have $n^{2-c} < \frac{1}{n^{c-2}}$, which goes to 0 as n grows. Thus the fullest bin has $O\frac{\lg n}{(\lg \lg n)}$ balls WHP.

6.3 Practical Implications

In load balancing applications, this result is powerful: even with purely random routing (which requires no centralized coordination), the load is nearly balanced. The maximum load is only $O(\frac{\lg n}{\lg \lg n})$, while the average load is 1. This is far superior to naive strategies that route to the least-loaded server, which would require expensive updates to a centralized load table.

7 Summary

The balls-and-bins model provides an elegant framework for understanding randomized load balancing and hashing. Key takeaways:

- Expected load in each bin is $\Theta(1)$
- Maximum load WHP is $O(\frac{\lg n}{\lg \lg n})$

References

[1] Michael Mitzenmacher, Eli Upfal. Probability and Computing: Randomized Algorithms and Probabilistic Analysis. Cambridge University Press, 2005.