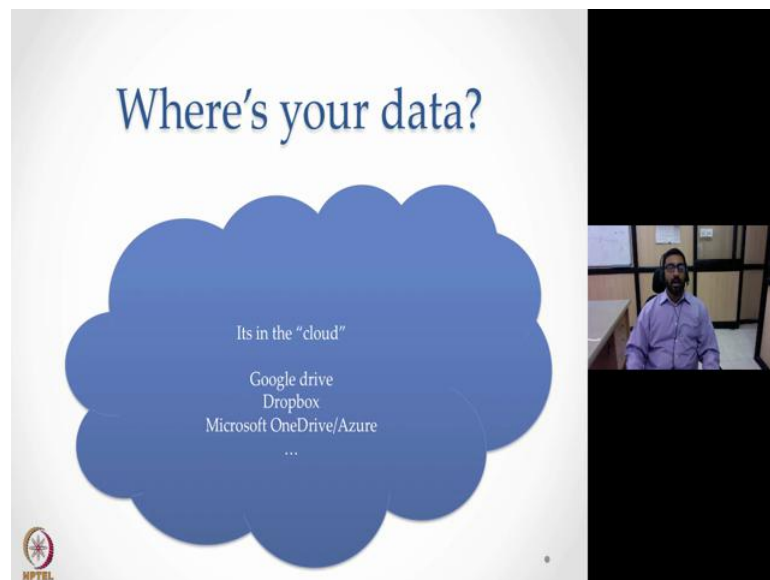


Algorithms for Big Data
Prof. John Ebenezer Augustine
Department of Computer Science and Engineering
Indian Institute of Technology, Madras

Week - 04
Lecture – 23
Overview

Hello everybody. Welcome to week 4, where will be continuing our discussion on Algorithms for Big Data. In this week, we will be particularly looking at Hashing and Pairwise Independence.

(Refer Slide Time: 00:30)



Now, in today's context when you think of data, where is it? Lot of us are storing it in the cloud, lot of us have a data in Google drive or Dropbox or Microsoft OneDrive, and we are getting use to running applications in the cloud. And what exactly does that mean? Essentially means that there is some large data center somewhere with lots and lots of servers and your data is stored in one or few of the service in this large data center.

(Refer Slide Time: 01:13)

How to ensure quick and easy access?

- Stored in a distributed fashion
- Where to look for it?
- Hashing
- How to understand hashing? **Segment 1**
 - Balls in bins
- Many hashing techniques: We will study
 - Chain Hashing **Segment 2**
 - Bloom Filters **Segment 3**

NPTEL

And when you need to access your data you need to have access to this data center, but obviously the company is not going to give you quick access to the data center, it is going to give access to your data through an appropriate interface. But, in any case that interface will have to know exactly where to look for the data, which server has it and within that is going to be a data structure, whole thing data not just from you from most of the people.

And so all your data needs to be access very quickly despite the factor there is so much complexity behind the scenes, and how do you think that that happens in the solution happens to be a simple technique called Hashing. Hashing is essentially using a function to take any key value and map it to an address and that address then tells you I mean that address could be the address of a server where your data is found, or it could be the address within a data structure where your data is found.

In any case hashing as you can imagine can be use in a variety of different ways and variety of algorithm, employee hashing as a sort of are primitive. So now, in this week we are going to understand hashing and particular we going to understand it in a very elementary way using the models of balls and bins. And essentially balls and bins means that you have some number of bins and you have some number of balls and I mean you

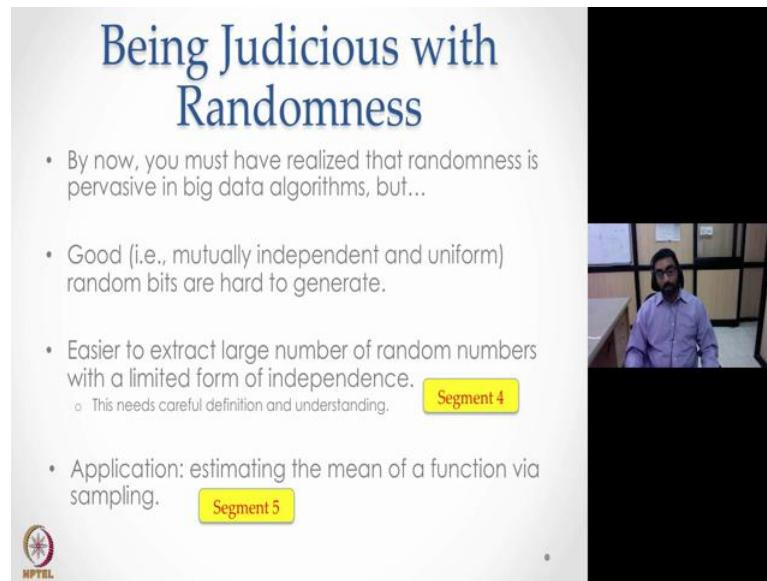
throw each ball uniformly at random and independently to 1 of the bins.

This is the experiment, and this has it turns out helps us to understand the hashing and you will see how that works. So, that is what we going to look at in the first segment of this week's lecture. Then we are going to look at two different shall we say data structures that employ hashing. One is something you might have already seen before chain hashing. We are going to try and understand it from the perspective of balls and bin, and we going to apply the balls and bin techniques to try and understand chain hashing. And, chain hashing is essentially a dictionary given key and the key value pair you can hash the key value into your data structure, find the particular location where to you can store the value and then later on retrieve it. It essentially is a dictionary ADT.

On the other hand, we will then continue to look at a new data structure potentially something which you not seen before call the bloom filter. A bloom filter is a slightly different simplified a dictionary some quite a dictionary. It is data structure that answers set membership queries, What that means is that it tries to maintain a set of items this could be the set of users in a large data center. Potentially the set of all possible user ID's is a very large universe and from that some number of used ID's are actually used by people as they log in to you say cloud computing system or something like that.


Now the question is, suppose someone newly wants to create a user ID, you want to know which user ID is already taken and which one is not taken. And this is essentially a set membership query. Bloom filter is helping you answer such queries even in a very compact way, even when the number of users is very large. So that will be the third segment of this week.

(Refer Slide Time: 05:18)



Being Judicious with Randomness

- By now, you must have realized that randomness is pervasive in big data algorithms, but...
- Good (i.e., mutually independent and uniform) random bits are hard to generate.
- Easier to extract large number of random numbers with a limited form of independence.
 - This needs careful definition and understanding. **Segment 4**
- Application: estimating the mean of a function via sampling. **Segment 5**



And then we will move on to the notion of pairwise independence. So just to give you context, finally you have realized that randomness is something that comes up in pretty much every big data algorithm. But one needs to realize that good what could we mean by mutually independent uniform random bits are actually hard to generate there is no easy clean way to do it. Essentially, these come from the fact that typically computers today or are deterministic machines. But as it turns out it is easy to extract large number of random numbers with a limited form of independence. We have already seen the definition of mutual independence.

In segment 4 we will see a more limited form of independence we will define what is called Pairwise independence or more generic k wise independence, and then we will try to understand the what we mean and what the implications of that definition are and as it turns out its easier to generate such random bits that have this limited independence.

And then, it is not of much value if you know just know how to generate them, but do not know how to apply them. So, what we will do is then we will apply these random bits. There are some this limited form of independence and will apply that to an application which is that of estimating the mean of a function via sampling. This is something that you might be able to imagine.

So, you would given a function, you want to estimate the mean, you do not want to look at every point of that function. As soon the function is not given an enclose form, but rather given the only way you can get the values of a function is by querying at a different locations. Now, in a situation like that sampling makes a lot of sense. What we will see here is how to use sampling using randomness that has this limited form of independence. And will see how we can effectively do that. So, this is that will be segment 5 in this week.

So with that, without further ado please enjoy the segments this week. And let me remind you that if you have any questions, if there is anything that is uncleared please drop a note in the forum and we will try to get back to you as quickly as we can.

Thank you.