

Lecture 12: Hashing, Distributed Hash Tables, and Chord

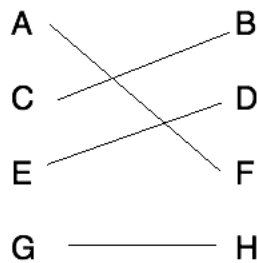
*Lecturer: Emery Berger**Scribe(s): Samuel Ginzburg*

12.1 Quick Overview of HashTables

HashTables are a data structure that seek to provide $O(1)$ access time on average to values given a specific key. HashTables use a hashing algorithm in order to hash the keys to an offset in an array, which can then be used to lookup the value associated with the key. Since arrays are of a fixed size, and HashTables grow in size over time, HashTables cannot be static, and must be dynamic.

The hashing function used to convert the keys into offsets for looking up the value in the array should generally form a random distribution if the algorithm is good.

Typically this results in a random projection from keys to values:



There is also the pigeonhole principle, which states that if a HashMap has M objects and N bins to place those objects in, and if $M > N$, then some bin has more than 1 item.

In addition to this we know that the expected length of the traversal is $E[\frac{M}{N}]$, where M is the number of objects, and N is the number of bins.

12.2 Distributed Hash Tables

The idea behind distributed hash tables is that the key-value pairs can be distributed onto multiple machines, to store particularly large hash tables.

12.3 Chord