Imani Palmer

6/10/13

Cuckoo Hashing

Cuckoo hashing is a programming technique used to resolve hash collision. The name of this hash collision method comes from a species of bird called the cuckoo. A young cuckoo moves other eggs out of the nest when it hatches. This is similar to the insertion of a new key into a cuckoo hashing table. When inserting a new key into a cuckoo hashing table there is a push of an older key into a different location. The main purpose of cuckoo hashing is that it represents the lookup and inserts runtime of perfect hashing. This means that cuckoo hashing guarantees O(1) lookup and O(1) insertion.

Cuckoo hashing was created in order to solve the desire of being able to look up items in constant time in the worst case. Cuckoo hashing requires that element *x* has two options of being stored. Instead of being stored at position *h1(x)*, it can be stored at either *h1(x)* or *h2(x)*. There is also one element that can be stored at any position. This means that there is no need to a data structure to hold collision elements; the lack of another data structure allows the look up of an item by checking two positions in an array. If both positions are occupied the insertion of a new element *x* is resolved by throwing out the current element *y* of position *h1(x)*. Then, check if the second position of element *y* is vacant. If vacant, place element *y* there. If not, then element *y* throws out element *x*. This continues until a vacant position is found or has taken too long. A new hash function will be chosen and the data structure is rehashed.

z

y

z

x

x

y

In the example above, is an illustrative representation of a successful insertion of key *x* by moving keys *y* and *z* from one table to a new table.

T1

T2

x

v

t

z

s

u

y

In the example above, the illustration demonstrates key *x* that cannot be properly inserted.

The advantages of cuckoo hashing involve two main concepts. The first and most ingenuous concept is that it improves on the worst case lookup time for hashing by making it constant. The second advantage is that it is simple to build and design. However, cuckoo hashing has many disadvantages. The disadvantage is if the inserted element has a very long path before insertion completes. A path is determined as long if an element cannot be placed within *O(log n)* moves. This only occurs when the load is very low which is rarely. Another disadvantage of the standard cuckoo hashing is that the failure rate is too high for many applications, even when there are multiple choices per element. The third disadvantage is that re-hashing is an expensive. Cuckoo hashing has various advantages and disadvantages, nonetheless, it accomplishes the main goal of being able to look up items in constant time in the worst case.

References

Hu, Alan. *Basic Algorithms and Data Structures*. University of British Columbia, 2010.

Mitzenmacher, Michael. *Cuckoo Hashing*. Cambridge: Harvard University, 2010.

Pagh, Rasmus. "Cuckoo Hashing for Undergraduates." *http://www.it-c.dk/people/pagh/papers/cuckoo-undergrad.pdf*. IT University of Copenhagen, 27 Mar. 2006.14 June 2013.

Pagh, Rasmus, and Flemming F. Rodler. "Cuckoo Hashing." *http://www.cs.tau.ac.il/~shanir/advanced-seminar-data-structures-2009/bib/pagh01cuckoo.pdf*. Department of Computer Science, Aarhus University. 14 June 2013.