Cairo University

Computer Engineering Dept.
Cairo University
Fall 2024

Advanced Database Systems
Lab 1
Hashing

Objectives

After the Lab, the student should be able to

- Implement different extendible external hashing.
- Understand complexity of managing directory and its size.

Introduction

Why Hashing? The amount of Data is increasing dramatically and the old data structure and way of storing is not longer good enough. Suppose we have a very large data set stored sequentially in a file. The amount of time required to lookup an element in the file is either O(log n) or O(n) based on whether the file is sorted or not. If the file is sorted then a technique such as binary search can be used to search. Otherwise, the file must be searched linearly. Either case may not be desirable if we need to process a very large data set. Therefore we discuss a new technique called hashing that allows us to update and retrieve any entry in constant time O(1). The constant time or O(1) performance means, the amount of time to perform the operation does not depend on data size n[1].

In database, we are mainly concerned with data in files not the Hash Table data structure, However the idea is quite the same. Our database stores a set of records and each record has a unique key (it should y3any :)). To speed up the insertion, update and retrieval we save the file using a hashing function.

[From Lecture Slides] Hashing for disk files is called External Hashing. The file blocks are divided into M equal-sized buckets, numbered bucket0, bucket1, ..., bucketM-1. Typically, a bucket corresponds to one (or a fixed number of) disk block. One of the file fields is designated to be the hash key of the file. The record with hash key value K is stored in bucket i, where i=h(K), and h is the hashing function. E.g., h(k)= K mod M Search is very efficient on the hash key. Collisions occur when a new record hashes to a bucket that is already full.

There are numerous methods for collision resolution like (Open addressing,Chaining,Multiple hashing), however when the collisions are too much it is time to resize the file. Resizing file can be made dynamically using extendible hashing, dynamic hashing and other algorithms. For the scope of this lab, we are going to implement extendible hashing.

Cairo University

Computer Engineering Dept.
Cairo University
Fall 2024

Advanced Database Systems
Lab 1
Hashing

Requirements

1) Using pen and paper, solve the following problem: using extendible hashing solve the following problem: A database is provided with record that contains two values (key, data), where the key is the search key we use to find the data in the table and hence we use it with the hashfunction. Both key and data are stored as one record in the database file. Perform the following operation in order using pen and paper:

```
insertItem(DataItem(13, 33)) //insert record (13,33)
deleteItem(13);  //delete item which has a key = 13
insertItem(DataItem(1, 20));
insertItem(DataItem(2, 70));
insertItem(DataItem(42, 11));
insertItem(DataItem(112, 54));
insertItem(DataItem(240, 16));
insertItem(DataItem(241, 99));
insertItem(DataItem(159, 30));
insertItem(DataItem(14, 45));
insertItem(DataItem(13, 88));
insertItem(DataItem(37, 1));
searchItem(13); //search for item which has key = 13
deleteItem(14);
deleteItem(13);
insertItem(DataItem(158, 5));
```

   Draw the directory and buckets after each step given that each bucket can hold 2 records only, Hashing function is (mod 256)

2) Continue the functions missing in ExtendibleHashing.cpp , Follow the TODOs and hints above each function it will help you.

Considerations

- Hashing function is implemented for you it is “mod (2^8)” then we start from the most significant bit.
- Printing functions are implemented for you.
- Code is written and tested under Linux and Windows using C/C++, feel free to use any.
- Cheating leads to **NEGATIVE GRADE** for both cheater and helper.
- Cheating from online resources is equivalently punished as the previous.
- Looking up syntax is not considered cheating, you are free to do so.