

Case Study on Hashing Applications

Yuvraj Nikam, Anuja Deshpande, Dev Bohra, Samiksha Dongre, Eeshan Umrani

Abstract:

The ongoing administrative process generates a lot of data. As a result, searching takes a while. Hashing techniques can speed up the search process and save time. By establishing references to the key that is transformed into the address in the database, hashing is a technique for directly accessing data in a table.

This paper conducts a thorough case study on the utilization of hashing techniques in Indonesia.

Keywords — Hash Table, Linear Quotient, Progressive Overflow

1. Introduction

The fourth most populated country in the world after China, India, and America is Indonesia, an archipelago of hundreds of large and tiny islands covering an area of around 2 million km². Data is being occupied by the expanding industry more and more.

Big Data is becoming more and more well-liked in practically all circles in Indonesia. In terms of governmental organizations, "Big Data" refers to a sizable data set that will be further examined or processed for purposes like forecasting, making decisions, and other things.

Because of the onset of the digital age and the subsequent increase in digital awareness, Big Data has been developing in Indonesia. Because Big Data can be used in various kinds of industries in Indonesia.

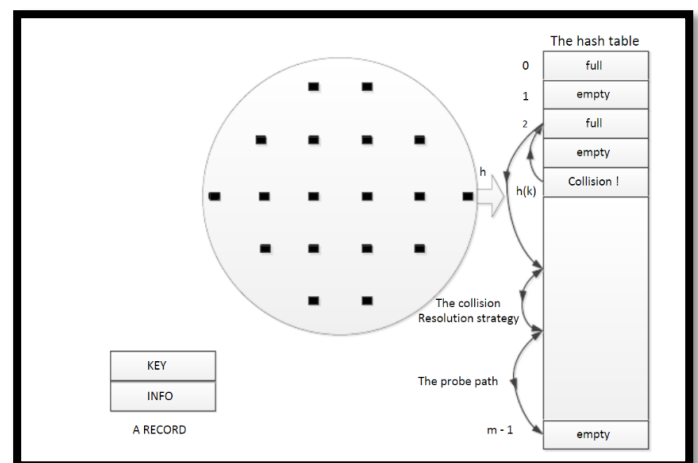
Data searching is a procedure that is used and necessary when building applications. While many algorithms are applicable, not all algorithms are very efficient when running. Small data might not make a big

difference, but as data creation continues, so does the growth of "Big Data." Big Data is an extremely big collection of data that can include text, numbers, or multimedia information that has been recorded in a specific database.

It takes a lot of time and work to manage an information technology database, including storing and searching data. To make management easier, database design is required.

It takes a while to process hundreds of thousands of data. Therefore, the investigation required extensive searching using hashing techniques. In order to conduct rapid, simple, and effective searches, search application interaction requires database structure consistency. Hashing algorithms can undoubtedly have a significant impact on how well searches perform across several databases.

2. Methodology



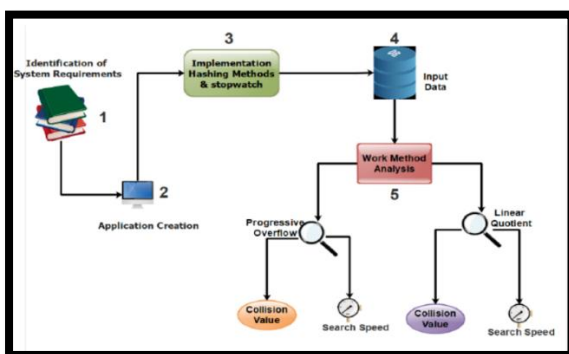
(Fig 2.1 H Hash function as mapping)

- The key K serves as a unique identifier for each R record. Additionally, K note R includes some relevant information that isn't mentioned in the INFO section. How to structure your notes to make it

easy to discover records in your table collection.

- Only the recording key is used to specify any retrieval requests or modifications.
- Keeping records in a table is a simple technique to establish organization.
- The table entry is either empty or contains a step.
- Examine all table entries in detail and search for records with a specific key.
- In a similar manner, a new record can be added to the table by looking for open spaces.

The location of the record containing the K key is shown in Figure 2.1 of hashing applying the transformation on key K. Assume the table has places or entries with the numbers 0, 1, ..., m-1. Then, by executing the function hash h, m maps all keys—which are presumptively very large—into the set (0, 1, ..., m-1) and describes it as a mapping. It can be claimed that the key K hashes to position s if $j(K) = s$. Yes, some keys could hash to the same place. Therefore, it is possible that the table h(K) entry is already filled by another key if you attempt to introduce a new key. In this situation, we require a system to search the rest of the database up until an empty entry.



(Fig 2.1 Research Flow)

Progressive overflow:

A traditional hash table implementation is linear probing or progressive overflow. A set of n keys is mapped into an array of m

sizes using the hash h function.

Linear Quotient:

In a hash table, the linear quotient or double hashing method is used to resolve collisions when two values create keys that are exactly the same. Another option for predicting commonly described objects across many datasets is double hashing.

The research flow diagram in Figure 2.2 shows numerous stages that are as follows:

1. In the recognized stage, the research difficulty is similar to choosing the right approach to apply in the case study. data that is required and how much of it the application will consume. Next, decide which system must be used to run the application for it to function properly.

2. The police officer logs in using their username and password. Officers can then continue to input employee data until system testing is complete.

3. Testing is carried out in two ways: calculating the average collision value and analyzing the hashing method using the initial hashing method incorporated into the program, which will later be used to see the accuracy of time.

4. The output in the form of a report, which will include each table of employee data, is the final stage.

5. The main house was used to implement the hashing process, which facilitated the use of performance analysis techniques. The button for each method and stopwatch function specifies the algorithm implementation process.

6. The data input process is split into two databases, the first of which has 1000 records and the second of which contains 2000 entries.

7. After selecting that the hashing method should be examined first, the analysis step

determines the key chosen at random. Each method produces two values, the first of which is the average collision value and the time value of the data search.

8. The last step compares the average value of collision and the amount of time spent searching using data from 1000 and 2000 records, making it easier to analyze the outcomes of each method's study.

3. Discussions

- A key is needed for the search process, which makes it quicker and more precise.
- In Indonesia, ID Employees is the key.
- There are 18-unit numbers in the key. Employee data is represented by the Indonesian ID Employee symbol. Therefore, it is possible to tell who owns ID Employees just by looking at it.
- A sound staffing management plan can address the issues mentioned above.
- Because each approach will produce a different average value of the collision and a different pace of search, the authors settled on the progressive overflow (PO) and linear quotient (LQ) hashing methods.
- Collision is the occurrence of key value collisions when they are placed on the same memory address table's index number.
- When calculating the average collision value without units and the amount of time spent conducting the search, utilize the stopwatch function with a time unit (millisecond).
- In case studies involving the use of memory management, both approaches were examined and found to be most beneficial.
- 2000 records total were utilized as the data to compare the effectiveness of the two approaches. If the performance of the two approaches was not produced by the 2000 record data, data would be added until the method appeared to be performing well.
- By using specified keywords, the

analysis process may rapidly and accurately search the data and show it.

4. Conclusion about the case study

- The Employee ID was used as the keyword in 15 tests, which examined 7 keys, with the average value of collision producing the same value on 7 of them and a different value on 8 of them. The number of collisions present while the address of the index is being determined affects the size of the average value of collision.
- Results of the experiment indicate that the linear quotient approach is superior to the progressive overflow method.
- The Employee ID was used as the keyword in 15 tests, which examined 7 keys, with the average value of collision producing the same value on 7 of them and a different value on 8 of them. The number of collisions present while the address of the index is being determined affects the size of the average value of collision.
- Results of the experiment indicate that the linear quotient approach is superior to the progressive overflow method.
- Therefore, it can be inferred that the hashing method may operate well on desktop programs, become a reference for the creation of the following method, and undoubtedly can be a reference.

5. Merits

- Hash tables have speed as their key benefit over other table data formats. When there are several entries, this benefit is more obvious (thousands or more).
- When the maximum number of entries can be known in advance, hash tables are very effective since the bucket array can be allocated once with the ideal size and never extended.

6. Demerits

- The cost of a decent hash function can be much higher than the inner loop of the lookup method for a sequential list or search tree, even though operations on a hash table often take constant time.
- As a result, hash tables are ineffective

when there are few entries.

- In general, hash tables have poor locality of reference, which means that the data they must retrieve is dispersed throughout memory in an apparently random manner.
- Hash tables can result in erratic access patterns, which can lead to microprocessor cache misses and lengthy delays.

7. Acknowledgement

We would like to thank our mentor, Yuvraj Nikam Sir, for timely guidance and assistance during the course of this Home Assignment.

References:

[1]https://www.researchgate.net/publication/328020140_Performance_Analysis_of_Hashing_Methods_on_the_Employment_of_App

[2]<http://rajaghoshtech2.blogspot.com/2010/03/advantages-and-disadvantages-of-hashing.html>

[3]<https://www.sciencedirect.com/topics/computer-science/hashing-algorithm>

[4] <https://www.quora.com/What-are-some-practical-applications-of-hashing>

[5] <https://levelup.gitconnected.com/the-3-applications-of-hash-functions-fab1a75f4d3d>

[6] <https://www.itperfection.com/network-security/about-hashing-and-its-applications-message-digest-sha1-sha2-sha3-md5-cryptography-encryption/>