Data Collision Prevention with Overflow Hashing Technique in Closed Hash Searching Process

Robbi Rahim1*, Nurjamiyah2 and Arie Rafika Dewi1

1Department of Health Information, Akademi Perekam Medik dan Infokes Imelda, Jl. Bilal Ujung Medan 20116, Indonesia
2School of Computer and Communication Engineering, Universiti Malaysia Perlis, Malaysia
3Department of Information System, Sekolah Tinggi Teknik Harapan, Jl. H.M. Joni No. 70 C, Medan 20216, Indonesia

*usurobbi85@zoho.com

Abstract. Hash search is a method that can be used for various search processes such as search engines, sorting, machine learning, neural network and so on, in the search process the possibility of collision data can happen and to prevent the occurrence of collision can be done in several ways one of them is to use Overflow technique, the use of this technique perform with varying length of data and this technique can prevent the occurrence of data collisions.

1. Introduction

The search process cannot separate from daily activities related to the information needs, the easiest to find information is to use certain keywords [1], and this search process would require an algorithm that has been tested in advance for the desired results accordingly.

Hash Search [2] is a search method that has a way of working similar to the method of direct search, which is by using certain formulas both at the time of placement and search data [3] [4]. Hash Search has better place space efficiency than Direct Search method [5]. The function (I - 1) used by the Direct Search method has a poor area of effectiveness, so the Hash Search method is added modulo operating function to cover the weakness of the direct search method. Modulo operation function is often applied to Hash Function, and the data shelter is called the Hash Table. The Hash function is not a one-on-one functionality as a function of the Direct Search method, so it is possible that some data have the same function results [5]. Collisions were occurring at the time of placement of data into tables so that a strategy is needed to overcome this collision. Strategies to overcome this there are various with advantages and disadvantages for each method.

This paper conducted search experiments with hash search method and overflow technique to avoid the occurrence of collision, the experiment is done with an extended dynamic and random data, the results are displayed with the use of overflow technique and without overflow technique, the results of the comparison shown can be an input how the search process is done and avoid collision data that could happen.
2. **Methodology**

Closed Hash [6] is introduced as the Hash way in which the data is directly stored in a Hash table with a specified table size that can be defined on its own [6]; in the application of the Hash function, it is possible that some data has the same result. The cause of occurrence at the time of placing data to be conducted search process [6] [7].

Strategies to overcome collisions by dividing the table into two pieces, namely the main table and overflow table with the provisions of the size of the main table is greater than the length of the table overflow [3]. The way of placement of data in the main table is done by using the Hash function. In the event of a collision, the data is placed on the overflow table. How to put data in the overflow table can be done sequentially or using the new Hash function. If the overflow table is full, then the data entered into the overflow table will be rejected, for the main table form and overflow can be seen as follows:

![Main Table and Overflow Table](image)

In the table above enter the number or value to be searched by using hash search algorithm and the search process starts from the new main table then to the overflow table.

3. **Proposed Method**

Experiments conducted with a data length of 15 pieces with the value 31,6,12,98,32,55,6,41,57,37,71,62,94,11,12 then the main table has 10 boxes and overflow table 7 boxes, to determine the position of the value in each box for the main table and overflow done calculation process as follows:

Data: 31, 6, 12, 98, 32, 55, 6, 41, 57, 37, 71, 62, 94, 11, 12

- **Data ‘31’**
  - Main table -> h(x) = 31 mod 10 = 1, Location 1 is empty so that ‘31’ data is placed at location 1
- **Data ‘6’**
  - Main table -> h(x) = 6 mod 10 = 6, Location 6 is empty so that ‘6’ data is placed at location 6
- **Data ‘12’**
  - Main table -> h(x) = 12 mod 10 = 2, Location 2 is empty so data ‘12’ is placed on location 2
- **Data ‘98’**
  - Main table -> h (x) = 98 mod 10 = 8, location 8 is empty so ‘98’ data is placed on 8 locations
- **Data ‘32’**
  - Main Table -> h (x) = 32 mod 10 = 2 -> collision with data ‘12’ the search continues to Overflow Table
  - Table overflow -> g (x) = 0, location 0 is empty so data ‘32’ is placed at location 0

The placement of the value 32 in the 2nd box main table cannot be performed because it already contains the data, to prevent the collision then the value 32 is inserted into the overflow table box starting from location 0, so the value of 32 is in the position 0 overflow table, the process is accomplished until all the value data is stored according to the main table and the overflow table, and generates the position of the numbers as follows:
After placing the values in the main table and overflow table is a technique to prevent collision data occurs, then the next process is to test to find the data contained in the main table and overflow, the process of searching the value in the overflow table is done sequentially from the initial position to end of overflow table, as an example of the value that wants to search is 37 of a set of values, then the process is to find the value into the main table with following functions:

Main Table -> h(x) = 37 mod 10 = 7 -> Data '37' not found, the search is continued into the Overflow Table

In the process of search overflow is done gradually with the following functions:

Overflow-> g(x) = 0 -> Data '37' not found.

Overflow-> g(x) = 1 -> Data '37' not found

Overflow-> g(x) = 1 -> Data '37' found on location 2

Experiments have been proved that collision could be bypassed and searching process with hash algorithm still can be done well, here is table test done with the amount of data which vary by using overflow technique and without technique of overflow, for data generated randomly.
Table 1. Searching Process Result

| No | Amount of Data | Without Overflow | With Overflow |
|----|----------------|------------------|---------------|
|    |                | Collision (ms)   | Search Time (ms) | Collision (ms) | Search Time (ms) |
| 1  | 25             | 3                | 0.43           | -              | 0.87            |
| 2  | 55             | 5                | 0.82           | -              | 1.01            |
| 3  | 100            | 4                | 1.11           | -              | 1.23            |
| 4  | 125            | 8                | 1.13           | -              | 1.42            |
| 5  | 175            | 8                | 1.19           | -              | 1.81            |
| 6  | 300            | 17               | 1.24           | -              | 2.62            |
| 7  | 500            | 49               | 2.46           | -              | 3.91            |
| 8  | 1000           | 67               | 5.76           | -              | 8.17            |
| 9  | 3000           | 81               | 7.31           | -              | 12.65           |
| 10 | 5000           | 262              | 8.67           | -              | 17.32           |

Table 1 showed the experiment results searching process with overflow technique and without overflow technique and obtained in the search process using the overflow technique obtained longer results.

Figure 7. Graph Comparison Result

The figure above shows varying time ranges in the data search process, and the result is the greater amount of data, then the gap search time is also higher, and the searching process without overflow technique much faster, but for the prevention of collision data overflow technique successfully done with the amount of data vary.

4. Conclusion

Overflow technique used to avoid the collision on data allocation, the overflow technique is useful when the table is smaller than the main table, besides also with applied its technique of overflow search process with hash algorithm no different result only process search so long compared Without using overflow technique.
References

[1] K. K. Soni, R. Vyas, and A. Sinhal, "Importance of String Matching in Real World Problems," *International Journal Of Engineering And Computer Science*, vol. 3, no. 6, pp. 6371-6375, 2014.

[2] Y. M. K. Omar, H. Osama, and A. Badr, "Double Hashing Sort Algorithm," *Computing in Science & Engineering*, vol. 19, no. 2, pp. 63-69, 2017.

[3] L. J. Guibas and E. Szemeredi, "The Analysis of Double Hashing," *Journal Of Computer And System Sciences*, vol. 16, pp. 226-274, 1978.

[4] D. Jia, W. Xie, and Y. Jiang, "Research of database query technology based on hash function," in *IEEE 8th International Conference on Information Science and Digital Content Technology (ICIDT)*, Jeju, South Korea, 2012.

[5] K. K. Pandey and N. Pradhan, "A Comparison and Selection on Basic Type of Searching Algorithm in Data Structure," *International Journal of Computer Science and Mobile Computing*, vol. 3, no. 7, pp. 751-758, 2014.

[6] J. N. V. V, K. M and K. A, "Frequent Itemset Generation using Double Hashing Technique," in *International Conference on Modelling Optimization and Computing*, 2012.

[7] E. Lodi and F. Luccio, "SPLIT SEQUENCE HASH SEARCH," *Information Processing Letters*, vol. 20, pp. 131-136, 1985.