2009 IEEE International Advance Computing Conference (IACC 2009)
Patiala, India, (

Choosing Best Hashing Strategies and Hash Functions

Mahima Singh, Deepak Garg
Computer science and Engineering Department, Thapar University ,Patiala.

Abstract

The paper gives the guideline to choose a best suitable hashing method hash function for a particular problem.
After studying the various problem we find some criteria has been found to predict the best hash method and hash function for that problem. We present six suitable various classes of hash functions in which most of the problems can find their solution.
Paper discusses about hashing and its various components which are involved in hashing and states the need of using hashing for faster data retrieval. Hashing methods were used in many different applications of computer science discipline. These applications are spread from spell checker, database management applications, symbol tables generated by loaders, assembler, and compilers.
There are various forms of hashing that are used in different problems of hashing like Dynamic hashing, Cryptographic hashing, Geometric hashing, Robust hashing, Bloom hash, String hashing. At the end we conclude which type of hash function is suitable for which kind of problem.

Keywords

Hashing, hash function, hash table , collision, resolution, buckets , cryptographic hashing, dynamic hashing.

1. Introduction

Hashing is the transformation of a string of characters into a usually shorter fixed-length value or key that represents the original string.
Hashing is used to locate and retrieve items in a database because its really do the work in a faster manner like to find the item using the shorter hashed key than to find it using the original value. It is also used in many encryption algorithms.[1][2][3]
Simply hashing is just opposite of sorting. Sorting arranges the records in some pattern. Hashing scatters the records throughout the hash table in a completely random manner. That's why hashing is appropriate for implementing a relationship among elements but it does not lend itself to operations which attempt to make use of any other relationships among the data. In implementing the hash table, other relationships may be destroyed.

To quickly locate a data record Hash functions are used with its given search key used in hash tables.
A hash table is a data structure that associates keys with values. The basic operation is to supports efficiently (student name) and find the corresponding value (e.g. that students enrolment no.). It is done by transforming the key using the hash function into a hash, a number that is used as an index in an array to locate the desired location (called bucket) where the values should be.
Hash table is a data structure which divides all elements into equal-sized buckets to allow quick access to the elements. The hash function determines which bucket an element belongs in.

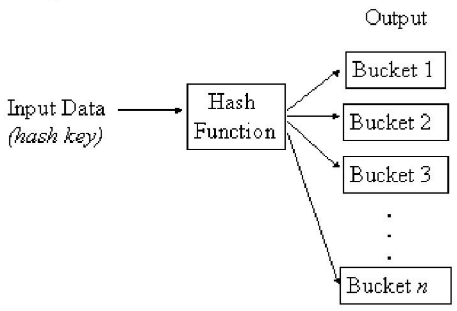

Fig1.Hash functions and hash keys

A Hash Table is a one-dimensional array indexed by an integer value computed by an index function called a hash function. Hash tables are sometimes referred to as scatter tables.

Typical hash table operations are:

- Initialization
- Insertion
- Retrieval
- Deletion

The advantage of hashing is that no index storage space is required, whereas inserting into other structures such as trees does normally require an index. Such an index could be in the form of a queue. In addition, hashing provides the advantage of rapid updates. And the disadvantage is that it usually lacks locality and sequential retrieval by key. Consequently, insertion and retrieval are more random. Another disadvantage is the inability to use duplicate keys. This is a problem when key values are very small (i.e. one or two digits).

Applications of Hashing

2.1 Database Indexing

Some database systems store their files into indexes.When a request comes for data, the key information is first found in the appropriate index file which references the exact record location of the data in the database file. The key information is sometimes stored as a hashed value in the index file.

2.2 Symbol tables

The tables used by compilers for maintaining the information about symbols from a program which access information about symbols very easily.

3.3 Data dictionaries

Data structures which support adding, deleting, and searching for data. the operations of a hash table and a data dictionary are similar, other data structures may be used to implement data dictionaries.

3.4 Network processing algorithm

Hash tables are basic objects of several network processing algorithms and applications, which include route lookup, packet classification and network monitoring.

3.5 Browser Cache.

Hash tables are used to implement browser caches.

Problems for Which Hash Tables are not suitable

- ➢ Problems for which data ordering is required.
- ➢ Problems having multidimensional data.
- ➢ Prefix searching especially if the keys are long and of variable-lengths.
- ➢ Problems that have dynamic data
- ➢ Problems in which the data does not have unique keys.

2. Collision

When two keys hash to the same bucket or location, an alternate location must be determined because two records cannot be stored in the same location. So finding an alternate location is called collision resolution. A collision resolution technique assures future key lookup operations return the correct, respective records.

4.1 Collision Resolution

There are a number of collision resolution techniques.But main are:

- ➢ Chaining
- ➢ Open Addressing
  - linear probing
  - quadratic probing
  - double hashing

An efficient collision resolution strategy is an important part of any hash table.

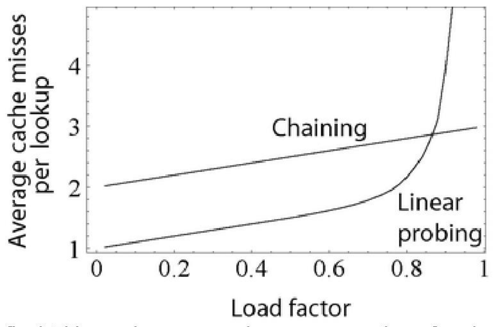

fig 2.This graph compares the average number of cache misses required to lookup elements in tables with chaining and linear probing. As the table passes the 80%-full mark, linear probing's performance drastically degrades.

3. Hashing Methods

Hashing is the process in order to obtain an index by chopping up the key and mixing it up in various ways which will be uniformly distributed over the range of indices hence the 'hashing'.So this section contains several common ways of doing this. [4] Some essential methods of hashing, or ways to insert values into a key accessed table are:

5.1 The Division Method

Algorithm: $H(x) = x \bmod m + 1$

Where m is some predetermined divisor integer (i.e., the table size), x is the preconditioned key and mod stands for modulo.Note that adding 1 is only necessary if the table starts at key 1 (if it starts at 0, the algorithm simplifies to ($H(x) = x \bmod m$).

In the applet, we did not add 1.

5.2. The Multiplication Method

This method is used in the applets. All the individual digits in the key are multiplied together, and after dividing the resulting number by the table size remainder is taken.

The algorithm in functional notation, is:

$H(x) = (a * b * c * d *....) \bmod m$

a, b, c, d, etc. are the individual digits of the item, and m is the table size, mod stands for modulo.

5.3. The Folding Method

Algorithm: $H(x) = (a + b + c) \bmod m$

a, b, and c represent the preconditioned key broken down into three parts,m is the table size, and mod stands for modulo.

In other words: the sum of three parts of the preconditioned key is divided by the table size.The remainder comes is the hash key.

5.4. Random-Number Generator

It is used for generating a pseudo-random numbers. Initially, the state of a chaotic system is digitized to form a binary string. To produce a second binary string this binary string is then

hashed and this becomes the second binary string which is used to seed a pseudo-random number generator. Now forming a password or cryptographic key for use in a security system The output from the pseudo-random number generator may be used in.
The algorithm must ensure that:
1.It always generates the same random value for a given key.
2.It is unlikely for two keys to yield the same random value.
The random number produced can be transformed to produce a valid hash value.

4. Various forms of Hashing

There is a vast variety of hash functions which are used in different kind of problems according to the need. The hash function(with the problems in which they are using these hash functions) which are presently in fashion or suits in presently needs of problems are Dynamic hash functions, Cryptographic hash functions, Geometric hash functions, Robust hash functions, Bloom hash functions, String hash functions. Hashing as a tool to associate one set or bulk of data with an identifier has many different forms of application in the real-world. Below are some of the more common uses of hash functions. [5]

Dynamic hashing

It is dynamic file access and provides the flexibility of handling dynamic tiles while preserving the fast access times expected from hashing. To support modern database systems this kind of fast, dynamic file access scheme is needed.

Its most outstanding feature is that any record can be retrieved in exactly one disk access.But it has a fixed retrieval speed and the storage utilization can be selected by the user. [6] [7]

Cryptographic Hashing

These are used in either in data or users verification and authentication like user's passwords and have the hash of the passwords stored on a system rather than having the password itself stored. A strong cryptographic hash function has the property of being very difficult to reverse the result of the hash and hence reproduce the original piece of data. Cryptographic hash functions are used to represent large quantities of data with a signal ID, they are also useful in data tamper and used in order to prove authenticity of a document via other cryptographic means. [8][9]Geometric Hashing

It is used in computer vision for matching geometric features against a database of such features, finds use in a number of other areas. When the recognizable database objects have undergone transformations or when only partial information is present even at that time matching is possible. For low polynomial complexity this technique is highly efficient .[10]

Robust Hashing

6.4.1 Robust Audio Hashing

It was made for précis a long audio signal into a concise signature sequence that can be used to identify the original record. The output sequence is denoted in the literature by alternate names, such as signature, fingerprint or perceptual hash values of the input that's called the robust audio hashing. The mapping tool of audio input to the signature is called perceptual hash function. Audio hashing methods were tested, to measure for robustness vis-à-vis non-malicious signal processing attacks on the one hand and to assess the uniqueness or randomness of the hash when audio files with different content.[11]

6.4.1 Robust image hashing

Basically it is related to cryptographic hash functions and it is sensitive only to perceptual change.Sign bit from the domain of Discrete Cosine Transform (DCT) s the basic concept for robust image hashing image robust hashing which is widely used in image processing and video processing, e.g. for compression or digital watermarking. The sign bit of this feature vector is extracted to form an intermediate hash. To derive a final hash the intermediate hash can be incorporated into some security mechanism. The advantages of the sign signal in DCT domain are verified in experiments evaluating robustness.

It is easy to be incorporated into image and video compression and watermarking due to the implementation in DCT domain.But subscriber cannot significantly alter some files without sacrificing the quality or utility of the data. This can be true of various files including image data, audio data, and computer code[12]

6.5 Bloom Filters

It allows for the state of existence of a very large set of possible type values to be represented with a much smaller piece of memory. Through the use of multiple distinct hash functions it can be achieved . And a by allowing the result of a query for the existance of a particular type to have a certain amount of error. by varying the size of the table used for the Bloom filter and also by varying the number of hash functions this error can be controlled.13]

String Hashing
String hashing is main operation, used in so many applications where fast access to distinct strings is required. For the data storage access and mainly within indexing of data and as a structural back end to associative containers *(i.e.: hash tables)* it is mostly used.[14]

Comparison of Linear probing v/s Double hashing

The choice between linear probing and double hashing depends primarily on the cost of computing the hash function and on the load factor of the table. Both use few probes but double hashing take more time because it hash to compare two hash functions for long keys.

8. Comparison of Open Addressing v/s Separate Chaining

It's somewhat complicated because we have to account the memory usage. Separate chaining uses extra memory for links. Although the Open Addressing extra memory implicitly within the table to terminate probe sequence. Open-addressed hash tables cannot be used if the data does not have unique keys. An alternative is use separate-chained hash tables.

Table 1: Comparison of Open Addressing Methods

Linear Probing	**Quadratic Probing**	**Double hashing**
Fastest among three	Easiest to implement and deploy	Makes more efficient use of memory
Use few probes	Uses extra memory for links and it does not probe all locations in the table.	Use few probes but take more time.
A problem occurs known as primary clustering	A problem occurs known as secondary clustering	More complicated to implement
Interval between probes is fixed often at 1.	Interval between probes increases proportional to the hash value.	Interval between probes is computed by another hash function

9. Best suited Hashing technique for a particular problem

	Problems solved by hashing	**Various types of Hashing used**
1.	For traditional databases or in access method. The basic purpose of access methods is to retrieve and update data efficiently. Especially when record has to be retrieved in exactly one disk access.[7]	Dynamic hash functions
2.	Password verification related applications. To authenticate a user, the password presented by the user is hashed and compared with the stored hash. This is sometimes referred to as one-way encryption.(8) For verification of message integrity. Determines whether any changes is to a message (or a file), by comparing message digests calculated before, and after, transmission .[15] For message digest for reliably identifying a file: several source code management systems, including Git, Mercurial and Monotone to uniquely identify them.[16] For security and performance reasons: most digital signature algorithms specify that only the digest of the message be	Cryptographic hash functions

	"signed", not the entire message. Hash functions can also	
3.	Problems like in computer graphics, computational geometry and many other disciplines. In these applications hashing function can be interpreted as a partition of that space into a grid of cells. [10] Problems of computer vision and structural alignment of proteins[18]	Geometric hash functions [Note: The technique is highly efficient and of low polynomial complexity.]
4.	Problems related with content-based retrieval, monitoring, and filtering[12] Problems of image processing and video processing, e.g. for compression or digital watermarking.[12] In authentication of both video data and still images and for integrity verification of visual multimedia [12]	Robust hash functions [Robust image hashing]
5.	Problems of most audio content identification systems[11] For identification of the original record: summarizing a long audio signal into a concise signature sequence, which can then be used for identifying.[11]	Robust hash functions [Robust Audio Hashing]
6.	Used in Web cache sharing: Collaborating Web caches use Bloom filters which make compact representations for the local set of cached files. Each cache periodically broadcasts its summary to all other members of the distributed cache. [19] Used in Query filtering and routing: The Secure wide-area Discovery Service, subsystem of Ninja project, organizes service providers in a hierarchy. Bloom filters are used as summaries for the set of services offered by a node. [20] Used in Compact representation of a differential file: a batch of database records are put in differential file contains to be updated. Used in Free text searching: Basically, the set of words that appear in a text is succinctly represented using a Bloom filter [21] For classification of various kinds of automobiles, for the purpose of re-detection in arbitrary scenes. The level of detection can be varied from just detecting a vehicle, to a particular model of vehicle, to a specific vehicle.	Bloom hash functions
7.	Mainly within indexing of data and as a structural back end to associative containers(i.e.: hash tables)[14]	String hash functions Used in the area of data storage access

So from above table it is seen that for the different problems we have choose a hash function which will be most fit to the problem and most efficient like in databases especially when record must be retrieved exactly in one access we must choose the dynamic hashing and for security and performance reasons like message integrity, password verification one should go with the cryptographic hashing, for image processing and for authentication for a video or a still image we go with the robust hashing, for problem like free text searching and redetection bloom hashing is good and for indexing of data string hashing will be most suitable.

10. Conclusions

In this paper we presented hashing methods/techniques and hash functions and compare them. On the basis of the study we compare the hashing collision resolving methods and find out when to use which technique. So the basis of that the conclusions in this paper, we summarize all the results and find the best suitable hashing methods for a particular problem.

11. References

[1] Introduction to Algorithms,2nd edition by Thomas H. Coremen, Charles E. Leiserson ,Ronald L. Stein,.PHI,Chapter 11.

[2] Algorithms in C, 2nd edition,3rd edition by Robert Sedgewick,. Addison- Wesley, 1998.Chapter 14.

[3] J. Zobel, S. Heinz, and H.E. Williams. In-memory Hash Tables for Accumulating Text Vocabularies.

[4] Experiments with universal hashing
Jyrki Katajainen and Michael Lykke
Department of Computer Science, University of Copenhagen Universitetsparken 1, DK-2100 Copenhagen East, Denmark.
11````````````````````````````````````

[5]Carter, J. L., And Wegmn, M. N. 1979. Universal classes of hash functions. J. Comput. Syst. Sci. 28, 2 (Apr.), 143-154.

[6] Dynamic Hashing Schemes
F L J. Enbody.Department of Computer Science, Michigan State University, East Lansing, Michigan 48823 H. CDU
Department of Computer Science, University of Minnesota, Minneapolis, Minnesota 55455

[7] Linear Hashing with Separators-A Dynamic Hashing Scheme Achieving One-Access Retrieval,University of Waterloo

[8] Keying Hash Functions for Message Authentication ,Mihir Bellare and Ran Canetti and Hugo Krawczyk.

[9] A Critical Look at Cryptographic Hash Function ,Scott Contini, Ron Steinfeld, Josef Pieprzyk, and Krystian Matusiewicz
Centre for Advanced Computing, Algorithms and Cryptography,Department of Computing, Macquarie University
[10]Geometric Hashing An Overview,Haimj.Wolfson Tel Aviv University,Isidore Rigoutsos
IBM T.F. Watson Research Center

[11] Robust audio hashing for Audio Identification,Hamza Özer, Bülent Sankur, Nasir Memon

[12] Image Robust Hashing based on DCT Sign,Longjiang Yu and Shenghe Sun
Dept. Automatic Test and Control, Harbin Institute of Technology, P.R.China

[13] Less Hashing, Same Performance: Building a Better Bloom Filter
Adam Kirsch_ and Michael Mitzenmacher__
Division of Engineering and Applied Sciences
Harvard University, Cambridge, MA 02138.

[14]Performance in Practice of String Hash Function,M V Ramakrishna,Justin Zobel
Department of computer science,RMIT.

[15] Cryptographic hash Function:An Overview Bart Preneell ESAT-COSIC Laboratory, K.U.Leuven K. Mercierlaan 94, B-3001 Leuven, Belgium.

[16]Cryptographic Algorithms and current trends

[17]On the Foundations of Modern Cryptography,Oded Goldreich,Department of Computer Science and Applied Mathematics
Weizmann Institute of Science, Rehovot, Israel

[18] Fast Protein Structure Alignment Algorithm Based on Local Geometric Similarity Chan-Yong Park, Sung-Hee Park, Dae-Hee Kil, Soo-Jun Park,Man-Kyu Sung, Hong-Ro Lee, Jung-Sub Shin, and Chi-Jung Hwang.

[19] E. Papapetrou, E. Pitoura, and K. Lillis, “Speeding –up Cache Lookups in Wireless Ad-Hoc Routing Using Bloom Filters”, the 16th Annual International Symposium on Personal Indoor and Mobile Radio Communications (PIMRC 2005), Sept 11-13, 2005, Berlin,Germany,2005.

[20] Andrei Broder, Michael Mitzenmacher, Network Applications of BloomFilters: A Survey, Internet Math 1(2003), no.4, 485-509.

[21] M.V. Ramakrishna. Practical Performance of Bloom Filters and Parallel
Free -Text Searching ,Communications of ACM,32 (10),October 1989