Fast Searching Technique on Sequential Databases in an Organization

[1]Samparthi V Suresh Kumar, [2]Kothapalli Chaitanya Deepthi, [3]Suryadevra Mohan Babu Chowdary

[1,2,3]Computer Science and Engineering Department, SirCRRCoE, Eluru

Abstract – We are in fast generation world. Everybody are dependent on Computer Systems. One or other way all countries in the world are dependent upon Computer Systems daily. There is lot of development in Computer hardware. Memory plays key role in a Computer System. We have memory in Tera Bytes. It may also go beyond that available memory. Do Computer Systems can retrieve old long years data? Yes, we can retrieve year's long data within couple of minutes/ seconds. Memory is incremental type. Every day bulk of data is added to it and becomes large databases. If we follow restrictions or assertions related to maintenance of Computer Software. Large data will be scattered around your disk(s), we can easily retrieve like data within few minutes/ seconds. We can easily solve this problem by maintaining index and binary search algorithms. It cost little bit of memory.

Keywords – **Large Database, Index, Binary Search, Computer Systems**

I. INTRODUCTION

Everyday we feed lot information to the Computer memory. Different types of memories are available like secondary storage in Computer Systems as shown in Fig. 1, Pen Drive, Laptops, Netbook, Tablets PC, Floppy Disk, Smart Phones, Microcomputers, Minicomputers, Supercomputers Mainframe Computers and Magnetic Tapes etc., various people use these memory Systems related to their work. They feed data to these Memories daily and they grow to large Databases.

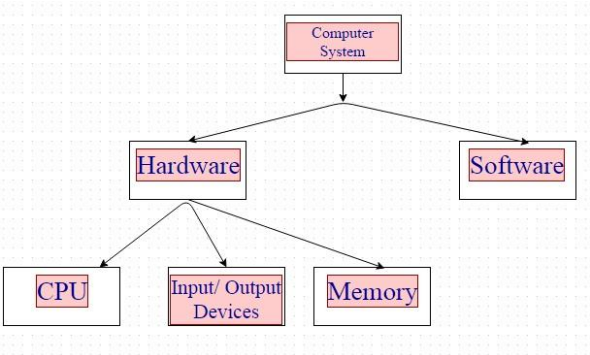

Fig. 1 Computer Memory

Fig. 2 Hard Disk with capacity 6TB.

Fig. 3 Magnetic Tape with capacity 185 TB.

Fig. 4 Pen Drive with capacity 128 GB.

Fig. 1 shows Computer System with hardware and software. We feed information to Computer System will be available in memory. Memory is secondary staorage. Maximum capacity of a hard disk is 6 Tera Bytes is shown in Fig. 2, a Magnetic Tape storage technology with capacity of 185 TB is shown in Fig. 3, and Pen Drive capacity of 128 GB is shown in Fig. 4.

II. METHODOLOGY

Everyday what we feed to any system will reside in memory. Memory is limited, memory is not unlimited. One or other day data will be filled based on the data feed to the systems. Mostly Computer Systems are used for business. There will be data available more but there should be a software which should provide data whenever required very fast.

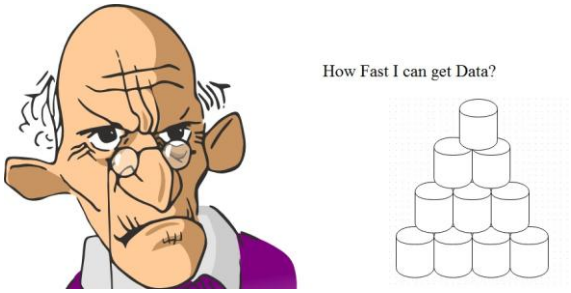

Fig. 5 How fast we can get data?

We are data rich, but information poor. The information store in a database will turn to data tomb after filling. Data tombs will increase timely based on feeding data. Per year there can be minimum of 2 disks for normal business. If supermarket or hypermarket the disks may range to 5 to 10. If we add slave disk to master disk then system will be slow. Connection is shown in Fig. 6. It has to go from one disk to another disk. It takes times to display interface with memory.

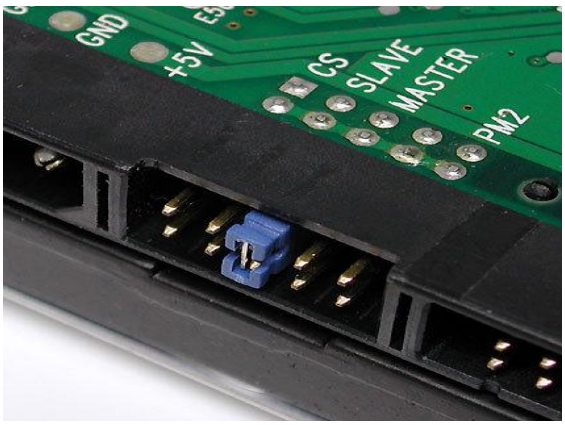

Fig. 6 Master and Slave connection to a memory.

Suppose there is a doctor who feeds old and new patient details to the Computer System. He fed patient disease and scan images related to the patient everyday. There may be regular visitors and sometimes there can be non-regular patients. Patients can be anyone, when they approach doctor the details of the patients should be hosted very fast. The software design should be like that they get the patient details very fast. X is a patient who approached doctor in the year 1990 for a disease and the same patient approached in the year 2016. That means the gap between these two years is 26 years. Now doctor have more than 10 disk(s) available with him. The patient details may be in 2nd disk and doctor may be using 10th disk. In real time there may me more disk(s), for our example purpose we are taking few disk(s).

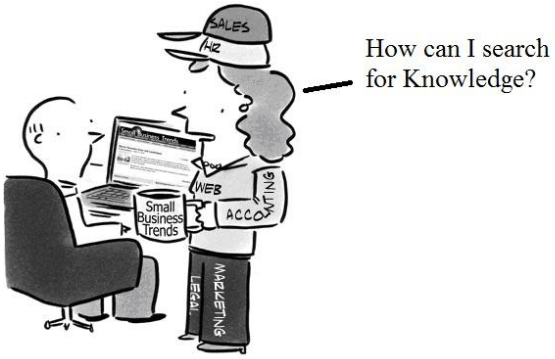

Fig. 7 Searching for Knowledge.

Searching process will take more time because it has to go for each location and every key data. Suppose the search is based on name means it will take hours. Sometimes multiple records may be displayed. Whenever doctor provides patient id entity based on patient details hardcopy and has to wait for sometime based on disk(s). the result of these queries may be multiple rows. Again the details should be proceeded in sequential manner by asking father name, mother name or mothers maid name. this process may take long time, sometimes hours, weeks etc.,

III. EXISTING SYSTEM

Old system is based on files. For every entry in reception, receptionist place one folder for each patient. Patient will carry folder along with him. Doctor enters details of the patient and his prescription details into the folder. From exit of Doctor, patient will submit that folder in reception to receptionist. They are placed in chronological order. Sometimes folders may be missed placed or lost, searching of the folders also will take long time. Important issue here is they are not safe. There can be overwriting of the documents which are harmful for patients.

Fig. 8 Files arranged in chronological order.

IV. PROPOSED SYSTEM

Create tables with normalization rules. Every table should have a primary key. The primary key should tag

with day and token number. Example 14022016045. 14022016 is current date 14/02/2016 and token number is 045. The arrangement of this primary key should be in linked list as given below:

14022016001 – 14022016002 – 14022016003 ……

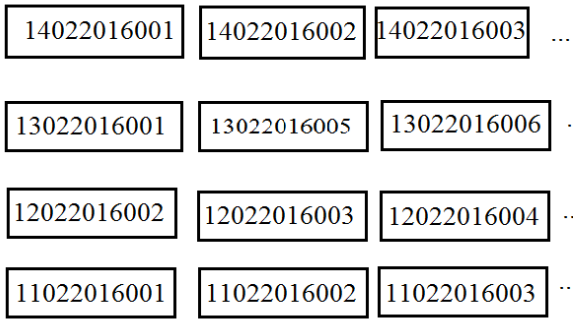

Fig. 9 Patient details (Indexed based)

Each patient should be allotted with fixed memory. Where the memory can be extended. The Fig. 8 arrangement of patient details should be in a index form as shown in Fig. 9.

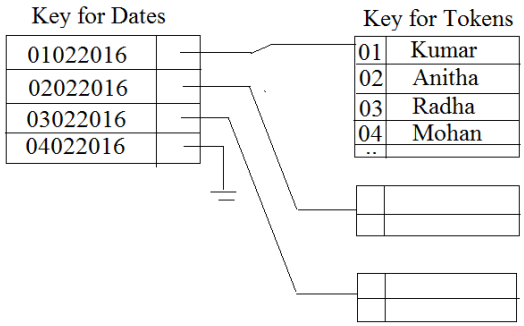

Fig. 9 Index Form

Search process may take place by providing key. After getting input key data it is separated to two keys one date and one for token.

I_n = Key1 + Key2

Input = 14022016045

Key1=14022016

Key2=045

Based on Key1 date is to be binary searched first. Next Key2 time to binary search. Binary search or half-interval search algorithm is used to get target patient details. The binary search can be classified as a dichotomic divide-and-conquer search algorithm and executes in logarithmic time. Storing of this type of data will be in chronology order. No need to sort again.

Algorithm

1. Let min=0 and max=n-1.

2. If max<min, then stop: target is not present in array. Return – 1.

3. Compare guess as the average of max and min, rounded down (so that it is an integer).

4. If array[guess] equals target, then stop. You found it. Return guess.

5. If the given was too low, that is, array[guess]<target, then set min=guess + 1.

6. Otherwise, the given was too high. Set max=guess – 1.

7. Go back to step 2.

Procedure

1. Sorted array: L1=[01022016, 02022016, 03022016, 04022016, 05022016].

2. Target value: X=04022016.

3. Compare X to 03022016 is greater. Repeat with L1=[04022016, 05022016].

4. Compare X to 04022016, equal, so go to the location.

5. Get the token number: T=10.

6. Sorted array of tokens: L2=[01,02,03,…09,10,..15].

7. Compare T to 8 is greater, Repeat with L2=[09,10,11,12,13,14,15].

8. Compare T to 12, Smaller, Repeat with L2=[09,10,11].

9. Compare T to 10, equal, so got to the location.

Implementation

A straight forward implementation of binary search is recursive.

```
int binary_search(int A[ ], int key, int imin, int imax)
{
if(imax < imin)
    return KEY_NOT_FOUND;
else
    {
    int imid=midpoint(imin, imax);
    if (A[imid] > key)
    return binary_search(A, key, imin, imid – 1);
else if (A[imid] < Key)
    return binary_search(A, key, imid+1, imax);
else
    return imid;
    }
}
```

Iterative

```
int binary_search(int A[ ], int key, int imin, int imax)
{
    while (imin<=imax)
    {
     int imid=midpoint(imin, imax)
    if (A[imid]==key)
        return imid;
    else if (A[imid] < key)
        imin=imid + 1;
    else
        imax = imid – 1;
    }
    return KEY_NOT_FOUND;
}
```

Complexity of above coding for worse case performance is O(log n), best case performance is O(1), average case performance is O(log n) and worst case space is O(1).

Java Code

```
importjava.util.scanner;
Class BinarySearchExample
{
    public static void main(String args[])
    {
    int counter, num, item, array[ ], first, last, middle;
    Scanner input = new Scanner(System.in);
    System.out.println("Enter number of elements");
    num=input.nextInt();
    array=new int[num];
    System.out.println("Enter  " + num + " integers");
    for(counter=0; counter < num; counter++)
        array[counter]=input.nextInt();
    System.out.println("Enter the search value");
    item=input.nextInt();
    first=0;
    last =num -1;
    middle=(first+last)/2;
    while(first < = last)
    {
    if(array[middle]<item)
        first=middle+1;
    else if (array[middle]==item)
    {
    System.out.println(item + "found at location" +
                                (middle + 1) + ".");
    break;
    }
else
    {
    last=middle – 1;
    }
middle = (first + last)/2;
 }
if(first > last)
  System.out.println(item + " is not found\n");
}
```

V. CONCLUSION

Finally, technology is growing same way storage in memory is also growing. Growing memory may lead to mess up and slow down the entire system. Storage increases as days are passing. When we require archive data the system falls and delay time increases. To overcome such problems we have to go for better searching techniques. This paper elevates better searching technique for an organization. It takes very less time to extract information from databases, if we follow some axioms related to storage.

REFERENCES

[1] D. J. Beebe, "Signal conversion (Book style with paper title and editor)," in Biomedical Digital Signal Processing, W. J. Tompkins, Ed. Englewood Cliffs, NJ: Prentice-Hall, 1993, ch. 3, pp. 61–74.

[2] M. Akay, Time Frequency and Wavelets in Biomedical Signal Processing (Book style). Piscataway, NJ: IEEE Press, 1998, pp. 123–135.

[3] G. B. Gentili, V. Tesi, M. Linari, and M. Marsili, "A versatile microwave plethysmograph for the monitoring of physiological parameters (Periodical style)," IEEE Trans. Biomed. Eng., vol. 49, no. 10, pp. 1204–1210, Oct. 2002.

[4] V. Medina, R. Valdes, J. Azpiroz, and E. Sacristan, "Title of paper if known," unpublished.

[5] E. H. Miller, "A note on reflector arrays (Periodical style—Accepted for publication)," IEEE Trans. Antennas Propagat., in press.

[6] T. Menendez, S. Achenbach, W. Moshage, M. Flug, E. Beinder, A. Kollert, A. Bittel, and K. Bachmann, "Prenatal recording of fetal heart action with magneto cardiography" (in German), Zeitschrift für Kardiologie, vol. 87, no. 2, pp. 111–8, 1998.

[7] J. E. Monzon, "The cultural approach to telemedicine in Latin American homes (Published Conference Proceedings style)," in Proc. 3rd Conf. Information Technology Applications in Biomedicine, ITAB´00, Arlington, VA, pp. 50–53.

[8] F. A. Saunders, "Electrotactile sensory aids for the handicapped (Presented Conference Paper style)," presented at the 4th Annu. Meeting Biomedical Engineering Society, Los Angeles, CA, 1973.

[9] J. R. Boheki, "Adaptive AR model spectral parameters for monitoring neonatal EEG (Thesis or Dissertation style)," Ph.D. dissertation, Biomed. Eng. Program, Univ. Fed. Rio de Janeiro, Rio de Janeiro, Brazil, 2000.

[10] J. P. Wilkinson, "Nonlinear resonant circuit devices (Patent style)," U.S. Patent 3 624 12, July 16, 1990.