A Methodology of Organizing the Data Base Archive Information

Prof. Dr. Sc. Dimo Arnaudov
Eng. Jordan Peychev, Sparky Trading GMBH
Eng. Bisser Blajev, Scorpion Shipping Ltd

The paper with an important problem of organizing large Data Bases /DB/. When the information of the DB exedes millions of items, e.g. 10-20 mill., then the problem of the response time comes in to view. One of the methods of reducing the response time is to take down off line the information not important for the given time. This data forms the archive of the information.

Here a methodology is described to show the strategy of deciding which is the not important information of the DB. A special algorithm is designed which helps the work of newlifiing the active DB and constructing the files of the archive information.

It is well known that nowadays the information on the online DB ranges millions of items. But inspire of the big memory (i.e. terrabites) there always should be a capacity problem with the disc DB memory. This problem concerns the time of response. The problem can be solved with the removal a part of this information to the archive. We mean that the archive information is such as storage, stated on disc devices, which are off line of the Information Retrieval System (IRS).

The first step is creating a strategy, that shows which of the documents of the DB must be moved to the Archive. Usually they are the "old" documents. But such a decision could be wrong if we haven't had added some other criteria. It's well known, that an old document could be very important and it shouldn't be moved tc the archive. That's the reason

why there are some special criteria, which are followed by the actual document removal from the DB. It is a combination of the documents "age" and its access number for a certain period of time.

Let L_i (t) means the age of the document $\underline{i}$ at the moment $\underline{t}$, N_i (t_1, t_2) means its access number during a certain time period (t_1, t_2), the criteria for the removal of the document to the archive can be stated as follows:

The document with number $\underline{i}$ is removed to the archive in case if:

(1) L_i (t) > T or

(2) $X \leq L_i(t) \leq T$ and $N_i(t-Y, t) < K$

In this case T is a constant, which determines the border age of the document in the DB.

K is the access number to the document during the previous Y units of time, in case if this document has been at least X units of time in the DB ($X \geq Y \geq 0$).

The constant T must be big enough. It depends on the essence of the DB. The parameters K, X, Y must be dynamically chosen (i.e. for each moment (t) when the problem, whether the document must be moved to the archive or files. It also is connected with the shortage of space in the DB.

The criteria for the document's removal from the DB to the Archive may be stored in the following map.

The document with number $\underline{i}$ will be moved to the archive if its age is more than the border age T, or if it is in the DB-X units of time ($X \leq T$) and during the last Y units of time ($X \geq Y \geq 0$) the accesses to this document are less than K.

It is obvious that according to this criteria (1) all the documents with an age bigger than T, will be moved to the Archive. There is a possibility, that "old", but "precious" documents could be removed to the archive. This is the reason why we should add another restriction. If N_i (t - Y, t) $\geq$ K,

the document with number $\underline{i}$ remains in the DB, despite of the fact that $L_i(t) > T$. This additional restriction combined with (1), (2), determines the criteria for the documents removal to the archive. In this case the criteria can be stated as follows. The document with number I will be moved to the archive file if:

(3) $L_i(t) > T$ and $N_i(t-Y,t) < K$

or

(4) $X \leq L_i(t) \leq T$ and $N_i(t-Y, t) < K$

Now we can easily determine the criteria for moving the document from the Archive to the DB:

(5) $N_i\ (t-Y, t) \geq K$ and $L_i(t) \leq T$

(6) $N_i\ (t-Y, t) \geq K$

This means that the document with number I will be moved back to the DB, if during the last Y units of time the accesses to the document are more or equal to K and its age is not bigger than T, or if the accesses to the document for the last Y units are more or equal to K, without concerning its age.

If we have already stated the above criteria, we can solve the problem, concerning the optimal usage of the DB memory.

Each new document may "enter" the DB if there is a "free space". That's why when the DB is "full", some of the documents must be removed to the archive in order to have "free space" for the new documents. Using the above criteria we could do this.

Then (fig. 1) the number of documents, which are moved to the Archive, will depend on the document, which must be moved back from the archive to the DB.

Fig. 1

the documents which remained in the DB		P1	P2	...	Pm	

The number of the document is "q". In this case must be fulfilled

(7) $Q \geq p^* + q$

Hence for the optimal usage of the DB memory space we must determine min R, where

$R = Q - p^* - q$, and $R \geq 0$.

The number p^* depends on the intensity of the Input to the system. The number q depends on the parameters of the system (T, X, Y, K).

In order to find min R when p^* is already stated, we must determine X, Y, K in a way that the formula $Q(X,Y,K)-q(X,Y,K)-p^*$ is minimal and nonnegative.

In order to solve this problem we must find the minimum of

$R = Q(X,Y,K) - q(X,Y,K) - p^*$

Here $p^* = p_1+p_2+\ldots..p_n$, where p_1 is the number of the input new documents at the I moment (see fig. 1)

The dynamical methods of determing the X, Y, K and having the min K as a result can be stated in the following way. We suppose that the statistical date has the form of Table 1.

Table 1.

Number of the I-document	the age	date of the input of X	Y = 1	$N_1(t-Y,t)$ Y = 2	. . .	Y = s
1						
2						
. . .						
N						

Such a table is formed for all of the documents of the DB and the archive. The information from these tables is usually stored on disc-devices. The essence of the problem is the determining which one of the documents must be moved to the DB and which of them must "go" to the archive. This problem has two stages. On the first stage parameters X, Y, K (when p* and k are determined) are found. For these parameters R is minimal and nonnegative (see the algorithm on fig. 1)

At the second stage we are closing the appropriate documents using the already determined X, Y, K parameters. In order to do this it is necessary that the tables of the DB and the Archive have been retrieved twice. At the first stage we determine only the number of the documents, which must be moved.

In order to do this we must create two new tables (for the DB and for the archive). They have 101 lines and S-rows, where S is the maximum number of the discrete units used by the system. In the table of the DB each element shows the number of the documents which fulfill (3), (4), where K=LK/100 for L=0.1…100 and Y=1,2…..S

The algorithm is on fig.2, the table is called TABOY. In the table of the Archive each element shows the number of the documents, which fulfill (5), (6). The table is called TABAR. The algorithm is the same.

After determining the elements of TABOX and TABAR we must calculate

A (i, j) = TABOX (i, j) - p* - TABAR (i, j)

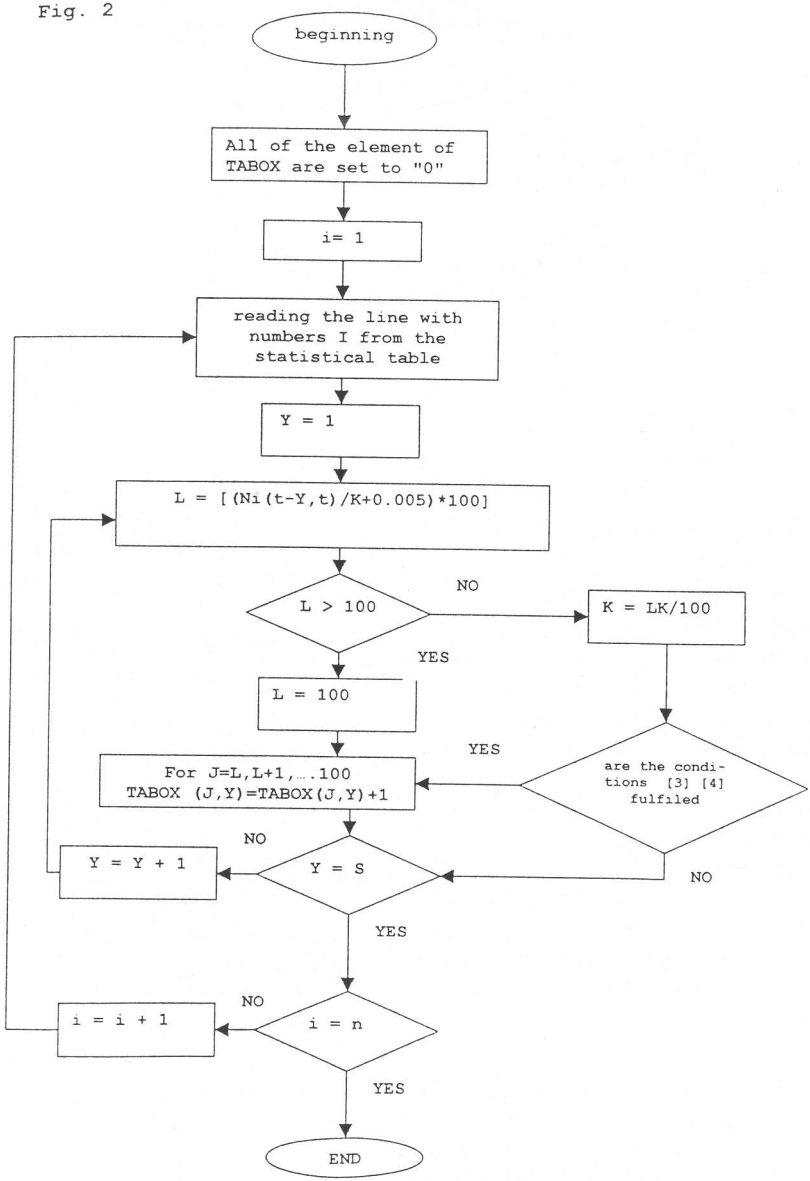
For j = 1,2….101; j = 1,2……. S

In order to find L and for which A (i, j) is minimal and nonnegative.

Then K = (LK)/100 and Y =

The described method gives the opportunity for maximum usage of the DB memory space capacity.

Fig. 2


beginning
All of the element of TABOX are set to "0"
i= 1
reading the line with numbers I from the statistical table
Y = 1
L = [(Ni(t-Y,t)/K+0.005)*100]
L > 100
NO
K = LK/100
YES
L = 100
YES
are the condi-tions [3] [4] fulfiled
For J=L,L+1,....100
TABOX (J,Y)=TABOX(J,Y)+1
NO
Y = Y + 1
Y = S
NO
YES
NO
i = i + 1
i = n
YES
END