REPLACEMENT ALGORITHMS BASING ON GENERALIZED MODELS FOR PROGRAM BEHAVIOUR

BY

OTTO SPANIOL

A 73 - 12

1. Introduction

Derivations of good replacement algorithms for virtual memory computers usually are based on intuitive considerations and on models which are either very restrictive or hard to realize.

King [6] for example uses the unrealistic assumption that the probabilities of referencing the pages of a program do not change during the execution of the program (cf. Aho, Denning and Ullman [1], o-order stationarity of programs). Various other approaches (including the very interesting "working set model") are described in papers of Denning [2 - 4].

Mostly discussion is restricted on demand paging strategies (only the page containing the references address is loaded into the fast memory unit if it is not yet present there). Thus the remaining problem is to determine the page which should be displaced from fast memory; this page must be reloaded into main memory if its contents have been changed during the residency of this page in fast memory. Furthermore in the cited papers the effects caused by program behaviour are neglected or insufficiently discussed.

In our opinion [7] was the first paper deriving a replacement algorithm on the base of two models of reference strings, namely the sequences of the addresses of referenced instructions and data respectively.

In the present paper results are derived for models of program behaviour which are more adequate to realistic situations than the models used in [7]. Furthermore a discussion of the new models is made if we are restricted on demand paging algorithms. Moreover, formulas are generalized on the case of not-Markowian conditions of program structure. Finally a replacement algorithm is derived which consists in a compromise between the strategy derived in [7] and customary strategies (e. g. LRU); by this modifications some difficulties are removed which are challenged by specific program behaviour which is not easy to express in formulas.

2. Summary of definitions and main results of [7]

A two-level storage hierarchy is considered where main memory (MM) and fast memory (FM) are divided into instruction memory I and data memory D.

A reference string is the sequence $x_{-r},\ldots,x_{-1},x,\ldots$ of instruction (or data) addresses a program needs during its execution.

Let $B_{-r},\ldots,B_{-1},B,\ldots$ be the corresponding pages in MM and let x be the address of the instruction (or data) actually referenced.

For the actual contents X of FM we assume $X \subset \bigcup_{i=-r}^{-1} B_i$ i. e. the FM only contains pages of the running program.

Now let $f^{-1}(y)$ be the MM-address of an instruction (or data) y in the FM.

Definition 1:

$z = z(x)$ is said to be the previous data address (PD) of $x \in$ MM if $|f^{-1}(z) - x| \leqq |f^{-1}(u) - x|$ for all $u \in X$.

If there are at least two addresses z with this property then take anyone of them as PD of x.

The difference $f^{-1}(z) - x$ is said to be a jump.

We assume jumps being independent random variables and distributions y^I and y^D (of the length of jumps in I and D respectively) being time-varying but maintaining some characteristic properties ($y_i \geqq y_{i+1}$ and $y_{-i} \geqq y_{-i-1}$ for $i \geqq 0$) during the execution of the program.

Furthermore as a Markow-condition we assume that the hitrate (probability of $x \in X$) depends on x and the PD z of x but not on previous other data.

Now let $B_1,\ldots,B_a$ be the pages a program needs for data (or instructions respectively) and assume that the FM consists of n pages ($n < a$) of size k then for the hitrate Q we have

$$Q = P(\ x \in X\ ) = \sum_{i=1}^{a} \sum_{m=1}^{a} w_{m,i} \cdot Q_{i-m,m} \cdot r_m$$

with the abbreviations:

$$w_{m,i} := P(\ B_i \in X \mid f^{-1}(x) \in B_m\ )$$

$$Q_{i,j} := P(\ f^{-1}(x) \in B_{j+i} \mid f^{-1}(z) \in B_j\ ;\ B_{j+i} \in X\ )$$

$$r_m := P(\ f^{-1}(z) \in B_m\ )$$

A replacement algorithm is said to be optimal if Q is maximized. Wanting to do so the replacement probabilities $w_{m,i}$ have to be choosen optimally. Since we are not restricted on demand paging strategies the derivation of optimal $w_{m,i}$ is very easy because in this case the rows of the matrix $W := (w_{m,i})$ can be considered as mutually independent.

Theorem 1 ([7]):

To maximize Q it is sufficient to maximize

$$Q' := \sum_{i=1}^{a} w_{m,i} \cdot Q_{i-m,m} \qquad (m = 1,\ldots,a)$$

by the constraints

$$\sum_{i=1}^{a} w_{m,i} = n \quad (m = 1,\ldots,a) \quad \text{and} \quad 0 \leqq w_{m,i} \leqq 1\ .$$

Hence Q' and Q are maximized if (given m) we assign 1 to the n probabilities $w_{m,i}$ belonging to the greatest numbers of the corresponding probabilities $Q_{i-m,m}$.

Theorem 2 ([7]):

Let $Q_{i,m} \geqq Q_{j,m}$ for $0 \leqq |i| < |j| \leqq a$.

Then for n odd we have:

$$W = (w_{m,i}) = \left[\begin{array}{cccccccccc}
1 & \cdots & 1 & & & & & & & \\
\vdots & & \vdots & & & & & & & \\
1 & & 1 & & & & & & & \\
0\,1 & & & 1 & & & & & & \\
 & 1 & & & 1 & & & & & \\
 & & \ddots & & & \ddots & & & & \\
 & & & 1 & & & 1 & & & \\
 & & & & 1 & & & 1\,0 & & \\
 & & & & & 1 & & 1 & & \\
 & & & & & \vdots & & \vdots & & \\
 & & & & & \underbrace{1 \cdots 1}_{n} & & & &
\end{array}\right] \begin{array}{l} \left.\right\} a \\ \left.\right\} \frac{n+1}{2} \end{array}$$

For n even there is a similar form of W, cf. [7].

The assumptions of theorem 2 are thoroughly discussed in [7].

As a result of this theorem we conclude that the hitrate is maximized if the neighbourhood of the PD of the actual data x is present in the FM.

We are not sure that the page containing x will be the page of the PD of the address x_1 referenced in the next step but in account of the characteristic properties of the distributions y^I and y^D and the Markow-condition this will be the best estimation of the page of the PD of x_1 (maximum likelihood).

That is why the best we can do is to load the neighbourhood of the referenced data address x, if the corresponding page is missed in FM. Pages of this neighbourhood which are already present in FM are obviously excluded from this replacement process.

3. Subdivision of MM and FM by address regions

In section 2, MM and FM are divided into two parts (for data and instructions respectively). This is caused by the different structure of the corresponding reference strings.

Testing the replacement algorithm described in [7] by means of simulations one remarks that the assumptions made for the data part D of MM are not realistic. This comes from the fact that D splits into different blocs or regions, each one corresponding to different data blocs. For example constants, simple variables, arrays and so on are stored in different (not necessarily coherent) regions of D.

That is why the application of the algorithm described in [7] leads to many superfluous replacement operations whenever the region of x doesnot coincide with the region of x_{-1}.

On the other hand for a suitable partition of D (and a corresponding partition of the FM belonging to D) into regions the assumptions made in [7] hold (locally) for each of these regions.

Definition 2:

Let $S^* := \{B_1, \ldots, B_a\} := \{b_1, \ldots\ldots, b_{ka}\}$ be the address space for data (or instructions) a program needs. Page B_i of size k is identified with its addresses $b_{(i-1)k+1}, \ldots, b_{ik}$.

$S \subset S^*$ is a <u>region</u> $\Leftrightarrow$ $S = \{B_i, \ldots, B_{i+j}\}$ $1 \leqq i < i+j \leqq a$

$\sum := \{S_1, \ldots, S_r\}$ is a partition of S^* into regions

$\Leftrightarrow \bigcup_{i=1}^{r} S_i = S^*$; $S_i \cap S_j = \emptyset$ ($i \neq j$)

The actual contents $X = \{X_1, \ldots, X_n\}$ at time t of the FM are given by a mapping

$f_t : S^* \longrightarrow X$

with $f_t(B_i) = \begin{cases} X_j & \text{if page } j \text{ of FM contains } B_i \\ \emptyset & \text{if } B_i \text{ is not present in FM at time } t \end{cases}$

The partition of S^* induces a partition of X:

$F_t : \sum \longrightarrow \mathcal{P}(X)$

$$\sum \ni S_i \longmapsto \bigcup_{B \in S_i} \{f_t(B)\}$$

If $|F_t(S_i)| \geqq 1$ (by $|H|$ we denote the number of pages of H)

then: $F_t(S_u) \cap F_t(S_v) = \emptyset$ $(u \neq v)$

$\bigcup_{u=1}^{r} F_t(S_u) = X$ (because X only contains data of the running program).

Definition 3: (PD relative to a region $S \in \sum$)

Let $x \in S \in \sum$ be the data actually referenced.

$z \in S$ is said to be PD of x in S, if:

1. $z \in F_t(S)$
2. $|f_t^{-1}(z) - x| \leqq |f_t^{-1}(u) - x|$ for all $u \in F_t(S)$

Let $n_i^{(t)} := |F_t(S_i)|$ $\qquad \sum_{i=1}^{r} n_i^{(t)} = n$

In the following sections we assume $n_i := n_i^{(t)}$ for all t, i. e. the partition of FM into parts corresponding to the the regions $S_1,\ldots,S_r$ is fixed. This assumption is discussed and weakened in the last section of the paper.

We are now ready to generalize the main results of [7] in two ways.

3. Generalization of the Markow-condition

Results similar to those derived in [7] are obtained if instead of the Markow-condition we assume that the hitrate depends on at most m data ($m \geq 1$ fixed) of the past.

Definition 4: (m-th PD of x)

z_m is said to be the m-th previous data (PD) of $x \in S \in \Sigma$ iff

1. z_j is PD of z_{j-1} in S ($j = 2,\ldots,i$)
2. z_1 is PD of x in S (cf. Def. 3).

In the following we assume that Q depends on at most m previous data $z_1,\ldots,z_m$.

Let $\mathcal{J}_m := (i_1,\ldots,i_m)$ $\qquad i_j \in [1:a]$ $\quad$ for $j = 1,\ldots,m$
and

$$w_{\mathcal{J}_m,i} := P(\, B_i \in F_t(S) \,|\, f_t^{-1}(z_1) \in B_{i_1} ; \;\ldots\; ; f_t^{-1}(z_m) \in B_{i_m} \,)$$

$$Q_{i,\mathcal{J}_m} := P(\, f_t^{-1}(x) \in B_i \,|\, B_i \in F_t(S) \;;\; f_t^{-1}(z_j) \in B_{i_j} \text{ for } j = 1,\ldots,m \,)$$

$$r_{\mathcal{J}_m} := P(\, f_t^{-1}(z_1) \in B_{i_1} \;;\; \ldots\ldots \;;\; f_t^{-1}(z_m) \in B_{i_m} \,)$$

Note that for $m = 1$ the definition of $Q_{i,\mathcal{J}_1}$ is slightly varied compared with the corresponding definition in section 2. This variation leads to a simplification of the following formulas.

Theorem 3:

Let $x \in S$ $\quad |S| = a \quad |F_t(S)| = n_S$

a. $$Q = \sum_{i,i_1,\dots,i_m=1}^{a} w_{\mathcal{I}_m,i} \cdot Q_{i,\mathcal{I}_m} \cdot r_{\mathcal{I}_m}$$

b. $$\sum_{i=1}^{a} w_{\mathcal{I}_m,i} = n_S \quad \text{for all } \mathcal{I}_m \in [1:a]^m \quad .$$

Proof:

$$\begin{aligned}
Q &= P(\, x \in F_t(S) \,) \\
&= \sum_{i=1}^{a} P(\, f_t^{-1}(x) \in B_i \text{ and } B_i \in F_t(S) \,) \\
&= \sum_{i=1}^{a} \sum_{i_1,\dots,i_m=1}^{a} P(\, f_t^{-1}(x) \in B_i \;;\; B_i \in F_t(S) \;; \\
&\qquad\qquad f_t^{-1}(z_1) \in B_{i_1} \;;\; \dots \;;\; f_t^{-1}(z_m) \in B_{i_m} \,) \\
&= \sum_{i=1,i_1,\dots,i_m=1}^{a} w_{\mathcal{I}_m,i} \cdot Q_{i,\mathcal{I}_m} \cdot r_{\mathcal{I}_m} \quad .
\end{aligned}$$

$$\begin{aligned}
\sum_{i=1}^{a} w_{\mathcal{I}_m,i} &= \sum_{i=1}^{a} P(\, B_i \in F_t(S) \mid f_t^{-1}(z_1) \in B_{i_1} \;;\; \dots \;;\; f_t^{-m}(z_m) \in B_{i_m} \,) \\
&= \sum_{i=1}^{a} \sum_{j=1}^{n_S} P(\, B_i \text{ contained in } C_j \in F_t(S) \mid \dots\dots \,) \\
&= \sum_{j=1}^{n_S} \sum_{i=1}^{a} P(\, F_t^{-1}(C_j) = B_i \mid \dots\dots \,) \\
&= \sum_{j=1}^{n_S} 1 \quad = n_S
\end{aligned}$$

Similar to the Markowian case (m=1) the hitrate and consequently the efficiency of the replacement algorithm is determined by the replacement probabilities $w_{\mathcal{I}_m,i}$.

Not being restricted on demand paging strategies we obtain (using the methods of section 2) the following result:

Theorem 4:

An optimal replacement algorithm is obtained by choosing the $w_{\mathcal{J}_m,i}$ (and-carrying out the resulting replacement operations) such that

$$Q' = \sum_{i=1}^{a} w_{\mathcal{J}_m,i} \cdot Q_{i,\mathcal{J}_m} \text{ is maximized } (\mathcal{J}_m \in [1:a]^m)$$

by the constraints

$$\sum_{i=1}^{a} w_{\mathcal{J}_m,i} = |F_t(S)| \quad \text{and} \quad 0 \leqq w_{\mathcal{J}_m,i} \leqq 1 .$$

Hence we have to assign 1 to those of the probabilities $w_{\mathcal{J}_m,i}$ which belong to the n_S greatest of the corresponding $Q_{i,\mathcal{J}_m}$ ($\mathcal{J}_m$ fixed) and 0 to the other $w_{\mathcal{J}_m,i}$ ($i \in \{1,\ldots,a\}$).

To apply this theorem we have to determine an adequate partition of the address spaces I and D in regions and, furthermore, to derive relations between the probabilities $Q_{i,\mathcal{J}_m}$ which are time-independent, generally valid and, last not least, simple (since the implementation of the resulting replacement algorithm should not be too expensive).

If $m = 1$ (i. e. the Markow-condition of section 2) the derivation of such relations is possible but here already these relations are sensitive to perturbations weakening the time-independence.

If $m > 1$ the derivation of simple and "stable" relations (i. e. insensitive against perturbations) seems to be hopeless.

That is why the rest of this paper is restricted again on the Markow-condition $m = 1$.

5. Analysis of demand paging algorithms

5.1. Introduction

When restricting on demand paging the methods developed in sections 2 and 4 are not any longer applicable since in any step (of time) at most one page may be displaced from the FM; for this reason the rows of the matrix $W = (w_{m,i})$ are by no means independent.

We want to maximize the following expression:

$$Q^{(t+1)} := \sum_{i=1}^{a} \sum_{m=1}^{a} w_{m,i}^{(t+1)} \cdot Q_{i-m,m}^{(t+1)} \cdot r_m^{(t+1)}$$

given $w_{m,i}^{(t)}$, $Q_{i-m,m}^{(t)}$ and $r_m^{(t)}$.
Hereby the superscript (t) indicates time or number of the step.

As in section 2 of the paper we assume that there are time-independent realations between the probabilities $Q_{i-m,m}$:

$$Q_{i,m}^{(t)} \geqq Q_{j,m}^{(t)} \Rightarrow Q_{i,m}^{(t+1)} \geqq Q_{j,m}^{(t+1)} \quad \text{for all i, m, j and t.}$$

The coefficients $r_m^{(t+1)}$ are influenced by the choice of $w_{m,i}^{(t+1)}$, i.e. by the replacement in step t.

In account of the relations holding between the $Q_{i,j}^{(t)}$ (cf. sect. 2), the Markow-condition and the fact that in the case of demand paging the replacement of at most one page in step t cannot change significantly the probabilities $r_m^{(t+1)}$ compared with the corresponding $r_m^{(t)}$ we may regard the values of $r_m^{(t)}$ as estimations of the numbers $r_m^{(t+1)}$; these estimations are choosen according to the maximum-likelihood principle.

Since $r_m^{(t)}$ is not influenced by a replacement in step t we try to solve the simplified problem

$$Q^{(t+1)*} := \sum_{i=1}^{a} \sum_{m=1}^{a} w_{m,i}^{(t+1)} \cdot Q_{i-m,m}^{(t+1)} \cdot r_m^{(t)} \overset{!}{=} \max_{w_{m,i}^{(t+1)}}$$

by the same constraints as in section 2.

In case of non demand paging algorithms the solution of the simplified problem could be received by maximizing the sums over the rows of the matrix W, but if we are restricted on demand paging $w_{m,i}^{(t+1)}$ depends on $w_{m,i}^{(t)}$ and on $w_{r,i}^{(t+1)}$ $(r \neq m)$.

5. 2. Relations between $w_{m,i}^{(t+1)}$ and $w_{m,i}^{(t)}$

Let $v \in S \in \Sigma$ be the page of the data referenced at time t. Then in a demand paging environment this page will be present in FM at time t+1, i. e. $w_{m,v}^{(t+1)} = 1$.

There are several cases to be discussed:

5.2.1. Let $w_{m,v}^{(t)} = 1$ for some $m \in \{1,\dots,a\}$.

Then $w_{m,i}^{(t)} = w_{m,i}^{(t+1)}$ for $i = 1,\dots,a$.

No replacements are carried out, row m of W remains unchanged.

5.2.2. Let $w_{m,v}^{(t)} = 1 - d_m < 1$ for some $m \in \{1,\dots,a\}$.

Let $\mathcal{I}_t := \{i_1,\dots,i_{n_S}\}$ be the set of page numbers of region S which are present in FM at time t.

5.2.2.1. Let $v \in \mathcal{I}_t$, i. e. no replacement at time t.

To guarantee $w_{m,v}^{(t+1)} = 1$ and $\sum_{i=1}^{a} w_{m,i}^{(t)} = \sum_{i=1}^{a} w_{m,i}^{(t+1)} = n_S$,

the probabilities $w_{m,j}^{(t+1)}$ must be multiplied by a factor $d > 0$

$$w_{m,i}^{(t+1)} = \begin{cases} 1 & \text{if } i = v \\ d \cdot w_{m,i}^{(t+1)} & \text{if } i \neq v \end{cases}$$

Obviously: $d = 1 - \dfrac{d_m}{n_S - 1 + d_m}$

5.2.2.2. $v \notin \mathcal{I}_t$.

Only in this case $Q^{(t+1)*}$ may be improved by an appropriate choice of the page which is to be replaced in favour of v.

Whitout knowledge of the criteria of the replacement algorithm no new relations between $w_{m,i}^{(t+1)}$ and $w_{m,i}^{(t)}$ $(i \neq v)$ are obtained. This is explained by the following discussion for two important classes of replacement algorithms.

A. Random replacement

Let $w_{m,i}^{(t)} = \begin{cases} 1 & \text{if } i \in \mathcal{J}_t \\ 0 & \text{if } i \notin \mathcal{J}_t \end{cases}$ and let $x \in v \notin \mathcal{J}_t$.

If the displaced page is choosen at random

then $$w_{m,i}^{(t+1)} = \begin{cases} 1 & \text{if } i = v \\ (n_s-1)/n_S & \text{if } i \in \mathcal{J}_t \\ 0 & \text{if } i \notin \mathcal{J}_t \cup \{v\} \end{cases}$$

B. t-deterministic replacement algorithms

Definition 5:

A demand paging replacement algorithm is said to be t-deterministic if and only if at time t the (number of the) page w to be replaced (if a replacement operation is necessary) is uniquely determined by the actual contents of the corresponding part of the FM, i. e. $w = f(\mathcal{J}_t)$.

For t-deterministic strategies we have

$w_{m,v}^{(t+1)} = 1$ and $w_{m,w}^{(t+1)} = w_{m,f(\mathcal{J}_t)}^{(t+1)} = 0.$

The other pages are not influenced by the replacement of v by w (concerning their presence in the FM) but the probabilities of presence in FM are changed; this is caused by the constraints $\sum_{i=1}^{a} w_{m,i}^{(t)} = \sum_{i=1}^{a} w_{m,i}^{(t+1)} = n_S$.

Therefore

$$w_{m,i}^{(t+1)} = \begin{cases} 1 & \text{if } i = v \\ 0 & \text{if } i = w \\ h \cdot w_{m,i}^{(t)} & \text{if } i \neq v,w \end{cases}$$

where $w_{m,v}^{(t)} = 1 - d_m < 1$, $w_{m,w}^{(t)} = e_m > 0$

and $$h = 1 - \frac{d_m - e_m}{n_S - 1 + (d_m - e_m)} .$$

Hence $w_{m,i}^{(t+1)}$ decreases if $d_m > e_m$, increases, if $d_m < e_m$ and remains unchanged, if $d_m = e_m$ (as one presumes from an intuitive aspect).

5.3. Evaluation of $Q^{(t+1)*}$ for t-deterministic algorithms

Suppressing indices (t) we have

$$Q^{(t+1)*} = \sum_{m=1}^{a} r_m \cdot \sum_{i=1}^{a} w_{m,i}^{(t+1)} \cdot Q_{i-m,m} \text{ .}$$

Now let $c_1, c_2 \in \{i_1,\dots,i_{n_S}\}$ $(c_1 \neq c_2)$ be two possible choices of replaced pages.

Let $Q_{c_j}^*$ be the hitrate $Q^{(t+1)*}$ if c_j is displaced $(j = 1,2)$.

Then according to the sign of the expression $\overline{Q} := Q_{c_1}^* - Q_{c_2}^*$ we prefer to replace c_1 (if $\overline{Q} > 0$) or to replace c_2 (if $\overline{Q} < 0$).

For t-deterministic algorithms we have:

$$Q_{c_j}^* = \sum_{m=1}^{a} r_m \cdot \sum_{\substack{i=1 \\ i \neq c_j}}^{a} w_{m,i}^{(t+1)} \cdot Q_{i-m,m}$$

$$= \sum_{m=1}^{a} r_m \cdot \Big(Q_{v-m,m} + \sum_{\substack{i=1 \\ i \neq c_j, v}}^{a} w_{m,i}^{(t+1)} \cdot Q_{i-m,m}\Big)$$

$$= \sum_{m=1}^{a} r_m \cdot \Big(Q_{v-m,m} + h(c_j) \cdot \sum_{\substack{i=1 \\ i \neq c_j, v}}^{a} w_{m,i}^{(t)} \cdot Q_{i-m,m}\Big)$$

where $h(c_j) := 1 - \dfrac{1 - w_{m,v}^{(t)} - w_{m,c_j}^{(t)}}{n_S - w_{m,v}^{(t)} - w_{m,c_j}^{(t)}}$.

Hence

$$\overline{Q} = Q_{c_1}^* - Q_{c_2}^*$$

$$= \sum_{m=1}^{a} r_m \cdot \Big(h(c_1) \cdot \sum_{\substack{i=1 \\ i \neq c_1}}^{a} w_{m,i}^{(t)} \cdot Q_{i-m,m} - h(c_2) \cdot \sum_{\substack{i=1 \\ i \neq c_2}}^{a} w_{m,i}^{(t)} \cdot Q_{i-m,m} \Big)$$

Because of the great number of parameters a general determination of the sign of $\overline{Q}$ seems to be impossible.

In the following we discuss the simplifications we obtain for the class of strongly deterministic demand paging algorithms, a class containing most uf the usual demand paging strategies.

5.4. Strongly deterministic demand paging algorithms

Definition 6:

A demand paging algorithm is strongly deterministic

$$w_{m,i}^{(t)} = \begin{cases} 1 & \text{if } i \in \mathcal{J}_t \\ 0 & \text{if } i \notin \mathcal{J}_t \end{cases}$$

for all i, m, t,
for all regions S of MM
and for all corresponding parts $\mathcal{J}_t = \mathcal{J}_t(S)$ of FM.

Roughly spoken all algorithms which never displace at random are strongly deterministic.

For strongly deterministic algorithms we have the following obvious results:

a. The rows of $W = (w_{m,i}^{(t)})$ are equal.

b. Let $x \in v \notin \mathcal{J}_t$; let $w = f(\mathcal{J}_t)$ be the replaced page.

Then: $$w_{m,i}^{(t+1)} = \begin{cases} 1 & \text{if } i \in \mathcal{J}_t \cup \{v\} \setminus \{w\} \overset{\text{Def}}{=} \mathcal{J}_{t+1} \\ 0 & \text{if } i \notin \mathcal{J}_{t+1} \end{cases}$$

Now we establish a replacement criterion for strongly deterministic replacement algorithms:

Theorem 5:

For str. determ. alg. a maximum of $Q^{(t+1)*}$ is obtained as follows: If $x \in v \notin \mathcal{J}_t$ then displace $w \in \mathcal{J}_t$ where w is given by the condition

$$\sum_{m=1}^{a} r_m \cdot Q_{w-m,m} = \min \quad \text{for } w \in \mathcal{J}_t \,.$$

Proof:

Using the formulas of section 5.2. we see that 5.2.2.1. is impossible whereas for 5.2.2.2.B. we have $d_m = e_m = 1$, i. e. $h = h(c_j) = 1$,
whence

$$Q^{(t+1)*} = \sum_{m=1}^{a} r_m \cdot \sum_{i=1}^{a} w_{m,i}^{(t+1)} \cdot Q_{i-m,m}$$

$$= \sum_{m=1}^{a} r_m \cdot \sum_{i \in \mathcal{I}_t \cup \{v\} \setminus \{w\}} Q_{i-m,m}$$

$$Q_c^{*} = \sum_{m=1}^{a} r_m \cdot \sum_{i \in \mathcal{I}_t \cup \{v\} \setminus \{c\}} Q_{i-m,m}$$

$$= \sum_{m=1}^{a} r_m \cdot \Big(\sum_{i \in \mathcal{I}_t \cup \{v\}} Q_{i-m,m} - Q_{c-m,m} \Big)$$

That is why

$$Q_{c_1}^{*} - Q_{c_2}^{*} = \sum_{m=1}^{a} r_m \cdot (-Q_{c_1-m,m} + Q_{c_2-m,m})$$

and Q_c^{*} is maximized if $\sum_{m=1}^{a} r_m Q_{c-m,m}$ is minimized for $c \in \mathcal{I}_t$.

Even here a general decision of the sign of $\overline{Q}$ seems to be impossible by reason of the complicated structure of the r_i. But since the previous data (PD) relative to region S of the data $x_1 \in S$ referenced at time t+1 is likely to lie in page v or in another page of $F_t(S)$ belonging to the MM-neighbourhood of v and since we assume that the relations between the probabilities $Q_{j,m}^{(t)}$ hold independently of t we see that (to obtain a simple replacement criterion) we should displace from $F_t(S)$ the page which (in MM) has greatest distance of v.

This criterion is also motivated by the following informal discussion:

Let $r_v \gg r_i$ $(i \neq v)$. Then $Q_{c_1}^{*} - Q_{c_2}^{*} \approx r_v(-Q_{c_1-v,v} + Q_{c_2-v,v})$.

That is why $Q_{c_1}^{*} \geqq Q_{c_2}^{*} \Leftrightarrow Q_{c_2-v,v} \geqq Q_{c_1-v,v}$

This condition is satisfied if $|c_1 - v| \geqq |c_2 - v|$ (in MM).

If the maximum of the r_i is not r_v but r_u (where u is near to v) then perhaps it would be better to displace page $c \in \mathcal{J}_t$ which is given by $|c - u| \stackrel{!}{=} \max$ for $c \in \mathcal{J}_t$.

Now by reason of maximum likelihood r_u cannot by much greater than r_v; furthermore $|v - u|$ is small and the probabilities $Q_{j,k}$ are small too (for large j); for these reasons the displacement of a page c' which is determined by the condition $|c' - v| \stackrel{!}{=} \max$ for $c' \in \mathcal{J}_t$ invokes at most an insignificant reduction of the hitrate compared with the optimal value $Q^{(t+1)*}$.

6. Description of a realizable non-demand paging repl. algorithm

The algorithms derived in sections 2 - 5 have some properties weakening their applicability for real programs:

a. Slight perturbations of the assumptions may result in many superfluous replacement operations.

b. The automatic determination of a good partition of MM into regions for which jumps are distributed in accordance to the assumptions is difficult.

c. If a region S' is neglected (if for example $\overline{S} = S' \cup S''$ is regarded as a section which - to obtain a better partition - should be subdivided into two smaller regions S' and S'') the assumptions we made about program behaviour are by no means satisfied; many replacements are necessary whenever a data of S' is referenced and the PD of that data belongs to S'' and vice versa.

d. If a section S''' is idle during an interval $[t_1, t_2]$ $(t_1 \ll t_2)$ of time then by reason of the fixed partition of the FM into regions $F_t(S)$ (i.e. $|F_t(S)| = n_S$ for all t) the region $F_t(S''')$ is inefficiently used during $[t_1, t_2]$.

e. Since activity of regions changes by time we should extend our formulas to $d_t(S) := |F_t(S)| - |F_{t-1}(S)| \neq 0$, but what should we do to answer satisfactorily the following questions:

   1. Find conditions to determine times t and regions S for which $d_t(S) \neq 0$. (Nevertheless in the majority of times the partition of FM should remain unchanged).

2. If $d_t(S) > 0$ (i.e. S was very active in the past and this activity caused many page faults), which parts of other region(s) are to be displaced to enlarge $F_t(S)$ compared with $F_{t-1}(S)$?
3. If $d_t(S) < 0$, which parts of $F_{t-1}(S)$ are to be placed at the disposal of enlarging regions ?

By this reasons a fixed partition of FM into regions seems not to be a good policy.

On the other side the program behaviour of MM (principle of locality, dominance of small jumps) favours an organization of MM according to the algorithm derived in [7], i.e. loading a coherent part of a region S when a replacement takes place.

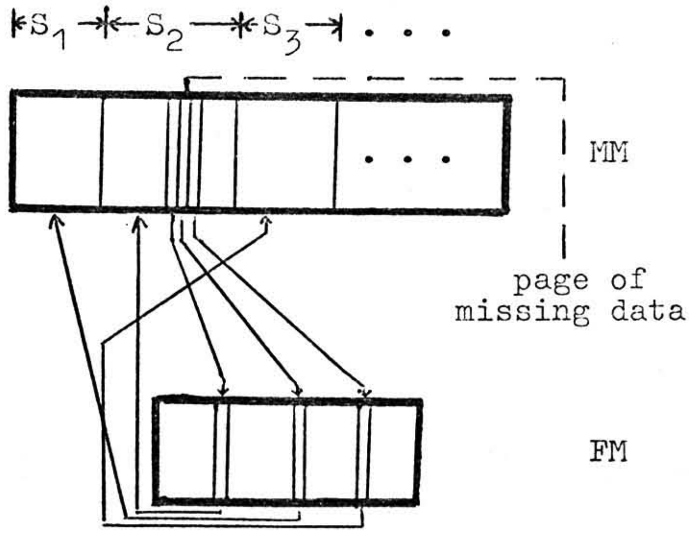
Since we recommend a new form of organization for FM the part of a region S which is contained in FM is not a coherent one and the number of pages which are to be loaded and replaced is not determined by first (in sections 2 - 5 this number was given by $F_t(S)$).

Hence even here an activity model could be introduced by which more pages are replaced for regions with greater activity and vice versa.

The simplest model (with regard of the implementation) is to load a neighbourhood of the missing page which always consists of a fixed number of pages.

The remaining problem is the organization of FM, i. e. the determination of pages which have to be displaced in favour of the incoming new pages:

a. A page already present in FM (if there is any) has not to be replaced.



Replacement algorithm proposed in this section. (Pages displaced from FM are determined by LRU e. g.).

b. The other pages (at least one for each replacement step) should be choosen by the criteria of a customary replacement algorithm (e. g. LRU).

Results of simulations show that this compromise between LRU and the replacement algorithm given in [7] is superior to usual strategies for most FORTRAN-like programs.

7. References

1 Aho, A. V., Denning, P. J. and Ullman, J. D.:
Principles of Optimal Page Replacement Algorithms.
J. ACM 18, 80 - 93 (1971)

2 Denning, P. J.:
The Working Set Model for Program Behaviour.
Comm. ACM 11, 323 - 333 (1968)

3 Denning, P. J.:
Virtual Memory.
Computing Surveys 2, 153 - 189 (1970)

4 Denning, P. J. and Schwartz, S. C.:
Properties of the Working Set Model.
Comm. ACM 15, 191 - 198 (1972)
Comm. ACM 16, 122 (1973)

5 Gelenbe, E.:
A unified Approach to the Evaluation of a Class of Replacement Algorithms.
IEEE Trans. Computers 22, 611 - 618 (1973)

6 King, W. F.:
Analysis of Demand Paging Algorithms.
IFIP 71, 485 - 490 (1972)

7 Spaniol, O.:
Optimale Nachladestrategien für Pufferspeicher unter Berücksichtigung der Programmstruktur.
Fachbereich Angewandte Mathematik und Informatik der Universität des Saarlandes; D - 6600 Saarbrücken
Bericht Nr. A 73 - 02.
(To appear in Acta Informatica).