The Properties and Limiting Behavior of
Working Set Size Strings and Flat-faults*

Tzong-yu Paul Lee

Computer Science Division
Department of Electrical Engineering and Computer Sciences
University of California, Berkeley

ABSTRACT

In the study of generative models based on working set size characterizations, working set size strings are used as inputs. The properties that an integer string should possess in order to guarantee the termination of reference string generation algorithms are discussed in this paper. A hierarchical view with respect to one window and two window working set size characterizations is presented. Working set size strings extracted from real traces may contain flat-faults which can cause difficulties in a class of reference string generation algorithms. The role and properties of flat-faults in the working set size strings are presented along with their limiting behavior under independent reference assumption. An upper bound of the flat-fault rate is also obtained in this paper.

1. Introduction

In the study of program behavior, various analytic and generative models have been proposed [Spir77a]. A class of generative models based on working set size characterizations was recently investigated by Ferrari [Ferr81a], Dutt [Dutt81a], and Lee [Lee82a]. This model uses *wss* (working set size) strings to generate artificial page reference strings for program behavior studies.

There are basically two ways to obtain *wss* strings. One is to produce them using a stochastic model [Fall81a] or some other description of a program's reference behavior. The other is to extract them from real program traces [Dutt81a] [Lee82a]. The first approach immediately raises the question of the necessary and sufficient conditions for the existence of at least one reference string corresponding to a set of integer strings. It is clear that, for a *wss* string extracted from a real trace, the existence of a reference string is guaranteed. The question is really relevant only for the artificially generated *wss* strings. This problem is discussed in the following sections.

The string generation algorithms proposed in the generative models studied so far are based on the flat-fault free assumption [Dutt81a] [Lee82a]. In essence, this class of reference string generation algorithms will not generate a reference to a new page when the working set size remains unchanged. As a result, the artificial reference strings generated are without flat-faults for the given set of window sizes. A flat-fault occurs at time t when a page fault at time t is accompanied by the simultaneous dropping out of a page from the working

*The research reported here has been supported by the National Science Foundation under grant MCS80-12900.

set. The working set size is unchanged for this kind of page fault. In most programs, the working set size remains unchanged for a very large fraction of the execution time; therefore, the reference string generation process will be unnecessarily flexible and will become computationally unmanageable if we intend to accommodate all possible flat-faults at various times when working set size remains unchanged. To justify further the elimination of flat-faults, it is shown in the later section that flat-faults occur rather infrequently with normal values for the parameter of the working set policy. The *wss* strings extracted from a real trace may contain flat-faults; therefore, their use in the reference string generation algorithms with the flat-fault free assumption is examined in the following sections.

2. Definitions

It is convenient to present here all the definitions needed later in this section. Some of the basic definitions, such as those of reference string and working set, are included for completeness. Explicit references to the parameter τ in many of the following definitions can be omitted if no ambiguity arises.

Definition 2.1

A reference string R is a sequence of references to a page set $P = \{1,2, \cdots ,n\}$. In essence, a reference string R, usually represented as $r_1 r_2 \cdots r_t \cdots$ where $r_t \in S$, is a mapping from the set of positive integers $I = \{1,2, \cdots \}$ to set P.

Definition 2.2

Given a reference string $R = r_1 r_2 \cdots r_t \cdots$ and a window size $\tau \in I$, the working set $W(t,\tau)$ at time t is the set of pages referenced in the interval $[max(1,t-\tau+1),t]$. That is, $W(t,\tau)=\{r_i \mid i \in [max(1,t-\tau+1),t]\}$. The working set size $w(t,\tau)$ at time t is the cardinality of $W(t,\tau)$, i.e., $w(t,\tau)=\mid W(t,\tau) \mid$.

Definition 2.3

The *wss* string $S(\tau)$ of a reference string R is a sequence of positive integer numbers $w(t,\tau)$, i.e., $S(\tau)=w(1,\tau),w(2,\tau), \cdots ,w(t,\tau), \cdots$

Definition 2.4

Given a reference string $R = r_1 r_2 \cdots r_t \cdots$ and a window size $\tau \in I$, a page fault occurs at time t if $r_t \notin W(t-1,\tau)$.

Definition 2.5

Given a reference string $R = r_1 r_2 \cdots r_t \cdots$ and a window size $\tau \in I$, a flat-fault occurs at time t if $w(t,\tau)=w(t-1,\tau)$ and $r_t \notin W(t-1,\tau)$.

In essence, a flat-fault occurs when there is a page fault at time t and there is also a page that drops out of the working set. The following definitions are based on those given in [Ferr81a], and on some of the results in [Ferr82a].

Definition 2.6

Given an integer bound τ, a string $S(\tau) = s_1 s_2 \cdots s_i \cdots$ of integers is called a *bpc* (bounded positive continuous) string if the following three conditions hold:

(i) $s_1=1$
(ii) $0<s_i \le \tau$ for $1 \le i$
(iii) $|s_i - s_{i-1}| \le 1$ for $2 \le i$

Definition 2.7

A *bpc* string $S = s_1 \cdots s_i \cdots$ with bound τ is *feasible* if for any prefix of length n of S there exists a reference string $R = r_1 \cdots r_n$ such that, with window size τ, the *wss* string of R coincides with the prefix of S in question.

Definition 2.8

Given a *bpc* string $S(\tau) = s_1 s_2 \cdots s_i \cdots$, the decrement count $d(\tau)_t$ at time t is the number of decrements in substring $s_t s_{t+1} \cdots s_{t+\tau-1}$. The number of decrements in this substring is the number of times that $s_i - 1 = s_{i+1}$ where $i \in [t, t+\tau-2]$.

Definition 2.9

A *bpc* string $S(\tau_l)$ is said to be greater than or equal to a *bpc* string $S(\tau_s)$ if $\tau_l > \tau_s$ and $s_t(\tau_l) \geq s_t(\tau_s)$ for all t's. We indicate this by the notation $S(\tau_l) \geq S(\tau_s)$.

Definition 2.10

A *bpc* string $S(\tau_l)$ is said to be *consistent* with a *bpc* string $S(\tau_s)$ if $S(\tau_l) \geq S(\tau_s)$, $d_t < s_t$ for both strings and all t's, and the following three properties hold :

(i). $s_t(\tau_l) = s_{t-1}(\tau_l) + 1$ implies $s_t(\tau_s) = s_{t-1}(\tau_s) + 1$ for all t's

(ii). $s_t(\tau_l) = s_{t-1}(\tau_l) - 1$ implies $s_{t-\tau_l+\tau_s}(\tau_s) = s_{t-\tau_l+\tau_s-1}(\tau_s) - 1$ for all t's

(iii). $s_t(\tau_s) = s_{t-1}(\tau_s) - 1$ and $s_{t+\tau_l-\tau_s}(\tau_l) \geq s_{t+\tau_l-\tau_s-1}(\tau_l)$ implies the existence of a unique $k \in (t, t+\tau_l-\tau_s]$ such that $s_k(\tau_s) = s_{k-1}(\tau_s) + 1$ and $s_k(\tau_l) \leq s_{k-1}(\tau_l)$ for all t's.

Definition 2.11

Reference strings R_1 and R_2 are said to be *wss-equivalent* with respect to τ if both have the same *wss* string for the given window size τ.

It is clear that the relation of wss-equivalence with respect to τ forms an equivalence class.

3. Properties of Working Set Size Strings

In this paper, we use the terms *single* τ and *double* τ *working set size characterization* to refer to *wss* characterizations of a given reference string based on one and two window sizes, respectively. The results presented here are divided into two categories accordingly. With a single τ working set size characterization, we call the single *wss* string $S(\tau)$. With a double τ working set size characterization, we call the two *wss* strings $S(\tau_l)$ and $S(\tau_s)$, respectively, where $\tau_l > \tau_s$.

3.1. Single τ Working Set Size Characterizations

With a single τ working set size characterization, we can partition the set of reference strings into two disjoint sets $\mathbf{R}_f(\tau)$ and $\mathbf{R}_n(\tau)$. $\mathbf{R}_f(\tau)$ is the set of strings that contain at least one flat-fault when processed with window size τ. $\mathbf{R}_n(\tau)$ is the complementary set of $\mathbf{R}_f(\tau)$. Two sets of *wss* strings are obtained from sets $\mathbf{R}_f(\tau)$ and $\mathbf{R}_n(\tau)$: $\mathbf{WSS}_f(\tau)$ and $\mathbf{WSS}_n(\tau)$. $\mathbf{WSS}_f(\tau)$ is the set of *wss* strings corresponding to the members of set $\mathbf{R}_f(\tau)$. $\mathbf{WSS}_n(\tau)$ is similarly defined. It is clear that there are usually more than one reference string having the same *wss* string with a single τ working set size characterization.

Furthermore, let us call the set of *bpc* strings with common bound τ $\mathbf{BPC}(\tau)$ and the set of *bpc* strings with common bound τ which also satisfy the constraint that $d_t < s_t$ for all t's $\mathbf{BPCD}(\tau)$.

Before we present and prove two propositions, a theorem by Ferrari [Ferr81a] is stated here without proof as Proposition 3.1. The statement of the original theorem is rephrased slightly in order to be consistent with the framework adopted in this paper.

Proposition 3.1

There exists a reference string $R \in \mathbf{R}_n(\tau)$ that has the *wss* characterization represented by $S(\tau)$ if and only if $S(\tau)$ is a bounded positive continuous string with bound parameter τ and $d_t < s_t$ for all t's.

Proposition 3.2

$\mathbf{WSS}_f(\tau)$ is a proper subset of $\mathbf{WSS}_n(\tau)$.

Proof : The proof is constructive. For any $S(\tau) \in \mathbf{WSS}_f(\tau)$ there is at least one reference string $R_0 \in \mathbf{R}_f(\tau)$ that is characterized by this *wss* string $S(\tau)$. The flat-faults associated with R_0 will be eliminated one by one in a series of successively constructed reference strings R_i's, with each R_i so constructed being wss-equivalent to R_{i-1}. Based on reference string R_{i-1}, reference string R_i is constructed as follows :

(i). Copy the reference string from R_{i-1} to R_i up to but not to include the first flat-fault at t in R_{i-1}. Assume that the page referenced at time t is x, and the page referenced at $t-\tau$ is y.
(ii). Copy the rest of the strings from R_{i-1} to R_i by interchanging the page names of x and y wherever they are referenced.

It is clear that reference string R_i has one fewer flat-fault than that of reference string R_{i-1}. Let us verify that R_i and R_{i-1} are wss-equivalent. From step (i), the working sets for both R_i and R_{i-1} for time $k<t$ are exactly the same. Thus, also working set sizes are the same. From step (ii), the working set sizes for both R_i and R_{i-1} for time $k \geq t+\tau-1$ are the same because only page names x and y are interchanged and none of the working sets covers any page referenced before time t. For time k such that $t \leq k < t+\tau-1$, the working sets $W(k,\tau)$ of both reference strings would be the same except possibly for pages x and y. Since page x will be in the working set for R_{i-1} in this range, we have two possible situations : y is not in the working set or y is in the working set but referenced only after time t because of the flat-fault assumption. In the former case, y is called x in the corresponding working set of R_i, and this will not affect the working set size. In the latter case, the page names x and y are interchanged at the same time, and hence the working sets are the same as well as the working set sizes.
Thus, given a fixed number of flat-faults m, because of the transitivity property of the wss-equivalence relation, there exists a flat-fault free reference string $R_m \in \mathbf{R}_n(\tau)$ with the same *wss* string $S(\tau)$.

The properness of the inclusion relationship can be shown by the example in Table 1. The reference string is constructed by repeating the sub-string of length 10.

Table 1 : $S(\tau)\in\mathbf{WSS}_n(\tau)$, $S(\tau)\notin\mathbf{WSS}_f(\tau)$										
time	1	2	3	4	5	6	7	8	9	10
$R\in\mathbf{R}_n(\tau)$	a	b	c	c	c	c	b	a	a	a
$S(\tau=4)$	1	2	3	3	2	1	2	3	3	2

The corresponding *wss* string is in the set $\mathbf{WSS}_n(\tau)$ because there is no flat-fault with a single τ characterization. The only times at which a flat-fault could occur are times 4 and 9, when the working set size remains unchanged. However, there is only one possible reference that can be generated at each of these times , i.e., that to the page referenced at time 3 and 7, respectively. If this is not so, we cannot decrement the working set size at later times as required by $S(\tau)$. Thus, this *wss* string $S(\tau)$ cannot be a member of $\mathbf{WSS}_f(\tau)$.

Q.E.D.

Proposition 3.3

The relationship among $\mathbf{BPC}(\tau)$, $\mathbf{BPCD}(\tau)$, and $\mathbf{WSS}_n(\tau)$ is

$$\mathbf{BPC}(\tau) \supset \mathbf{BPCD}(\tau) = \mathbf{WSS}_n(\tau)$$

where the inclusion relationship is proper.

Proof : The inclusion relationship is implied by the definitions of the two sets $\mathbf{BPC}(\tau)$ and $\mathbf{BPCD}(\tau)$. The properness of the inclusion relationship can be shown by the example in Table 2.

Table 2 : An Example of Working Set Size String (τ= 4)							
time	1	2	3	4	5	6	7
s_t	1	2	2	2	1	2	1
d_t	0	1	1	2	.	.	.

The string in the table is clearly a *bpc* string, but it is not in set $\mathbf{BPCD}(\tau)$ because at $t=4$, $d_t=s_t$.
The equality part can be shown as follows. If $x\in\mathbf{BPCD}(\tau)$, there exists a reference string R such that it has x as its *wss* string by Proposition 3.1. If reference string R has no flat-fault with a single τ characterization, then we have $x\in\mathbf{WSS}_n(\tau)$. Otherwise, $x\in\mathbf{WSS}_f(\tau)$ and, by Proposition 3.2, we still have $x\in\mathbf{WSS}_n(\tau)$. Thus, we have shown that $\mathbf{BPCD}(\tau)\subseteq\mathbf{WSS}_n(\tau)$. If $x\in\mathbf{WSS}_n(\tau)$, there is a reference string in $\mathbf{R}_n(\tau)$ such that it has x as its *wss* string. This reference string has its *wss* string satisfying the definition of $\mathbf{BPCD}(\tau)$ by the "only if" part result of Proposition 3.1. Thus, we have shown that $\mathbf{BPCD}(\tau)\supseteq\mathbf{WSS}_n(\tau)$. Combining the two, we have $\mathbf{BPCD}(\tau)=\mathbf{WSS}_n(\tau)$.

Q.E.D.

In summary, the necessary and sufficient conditions for an integer string to be a feasible *wss* string are stated in Proposition 3.1, i.e., the integer string is a *bpc* string with bound parameter τ and $d_t < s_t$ for all t's. By Proposition 3.2, the *wss* strings extracted from real program traces can be used as inputs to the class of flat-fault free string generation algorithms regardless of whether these *wss* strings contain flat-faults or not.

3.2. Double τ Working Set Size Characterizations

It may be wondered whether a flat-fault at time t processed with one working set window size implies a flat-fault at the same time when processed with a different working set window size. The following proposition answers such question.

Proposition 3.4

Given two window sizes τ_l and τ_s, where $\tau_l > \tau_s$, and a reference string R, there is no relationship between the presence of a flat-fault at time t with window size τ_l and the presence of a flat-fault at time t with window size τ_s.

Proof : This assertion can be proved by considering the string in Table 3 with τ_l=3 and τ_s=2. The flat-faults are indicated by asterisks in Table 3.

Table 3 : An Example of Flat-faults under double τ Characterization						
time	1	2	3	4	5	6
page	a	b	c	d	d	e
$S(\tau_l$=3)	1	2	3	3	2	2
flat-fault	.	.	.	*	.	*
$S(\tau_s$=2)	1	2	2	2	1	2
flat-fault	.	.	*	*	.	.

In this example, we can identify all possible combinations. At time 3, the τ_s-based characterization has a flat-fault, but not the τ_l-based characterization. At time 4, both characterizations have flat-faults. At time 5, neither characterization has a flat-fault. At time 6, the τ_l-based characterization has a flat-fault, but not the τ_s-based characterization.

Q.E.D.

Similar to the case of single τ working set size characterization, with double τ characterizations we partition the set of reference strings into two disjoint sets $\mathbf{R}_{nn}(\tau_l,\tau_s)$ and $\mathbf{R}_{ff}(\tau_l,\tau_s)$. $\mathbf{R}_{nn}(\tau_l,\tau_s)$ is the set of strings that contain no flat-fault with either (τ_l or τ_s) working set size characterization. $\mathbf{R}_{ff}(\tau_l,\tau_s)$ is the complementary set of $\mathbf{R}_{nn}(\tau_l,\tau_s)$. Two sets of *wss* strings are obtained from sets $\mathbf{R}_{nn}(\tau_l,\tau_s)$ and $\mathbf{R}_{ff}(\tau_l,\tau_s)$: $\mathbf{WSS}_{nn}(\tau_l,\tau_s)$ and $\mathbf{WSS}_{ff}(\tau_l,\tau_s)$. $\mathbf{WSS}_{nn}(\tau_l,\tau_s)$ is the set of *wss* string pairs corresponding to the members of set $\mathbf{R}_{nn}(\tau_l,\tau_s)$. $\mathbf{WSS}_{ff}(\tau_l,\tau_s)$ is similarly defined. It is clear that there is usually more than one reference string having a given pair of *wss* strings with two given window sizes.

Furthermore, let us define $\mathbf{BPCTT}(\tau_l,\tau_s)$ as the set of all *bpc* string pairs $(S(\tau_l),S(\tau_s))$ such that $S(\tau_l) \geq S(\tau_s))$ and $d_t < s_t$ for both strings and all t's. $\mathbf{BPCTTC}(\tau_l,\tau_s)$ is defined as the set of string pairs in $\mathbf{BPCTT}(\tau_l,\tau_s)$ such that $S(\tau_l)$ is consistent with $S(\tau_s)$.

Before we present and prove two propositions, a theorem by Ferrari [Ferr82a] is stated here without proof as Proposition 3.5.

Proposition 3.5

There exists a reference string $R \in \mathbf{R}_{nn}(\tau_l,\tau_s)$ that has the *wss* characterizations represented by $S(\tau_l)$ and $S(\tau_s)$ if and only if $S(\tau_l)$ is consistent with $S(\tau_s)$.

Unlike what happens with a single τ characterization (see Proposition 3.2), $\mathbf{WSS}_{ff}(\tau_l,\tau_s)$ is not a proper subset of $\mathbf{WSS}_{nn}(\tau_l,\tau_s)$, as we shall see in Proposition 3.6. Therefore, that $S(\tau_l)$ is consistent with $S(\tau_s)$ is only sufficient for the general class of string generation algorithms, i.e., the class of algorithms that may generate reference strings with flat-faults. Thus, *wss* string pairs extracted from real program traces need to be checked against this sufficient condition before they can be used as inputs to the class of flat-fault free string generation algorithms.

Proposition 3.6
The intersection of $\mathbf{WSS}_{ff}(\tau_l,\tau_s)$ and $\mathbf{WSS}_{nn}(\tau_l,\tau_s)$ is not empty, and neither set contains the other.

Proof : The proof can be obtained by considering three examples of string pairs. The first string pair (in Table 4) shows that the intersection is non-empty, because reference strings $R_1 \in \mathbf{R}_{nn}(\tau_l,\tau_s)$ and $R_2 \in \mathbf{R}_{ff}(\tau_l,\tau_s)$ both have the same *wss* string pairs. Notice that a flat-fault, indicated by an asterisk, occurs at time 6 for reference string R_2 with τ=4.

Table 4 : $(S(\tau_l),S(\tau_s)) \in \mathbf{WSS}_{nn}(\tau_l,\tau_s) \cap \mathbf{WSS}_{ff}(\tau_l,\tau_s)$							
time	1	2	3	4	5	6	7
R_1	a	b	c	c	c	b	a
R_2	a	b	c	c	c	a	b
$S(\tau_l=4)$	1	2	3	3	2	2^*	3
$S(\tau_s=3)$	1	2	3	2	1	2	3

The second string pair (in Table 5) shows that there is no reference string in $\mathbf{R}_{ff}(\tau_l,\tau_s)$ which can possess that *wss* string pair. The reference string is the same as that used in Table 1, and consists of repetitions of the sub-strings used in Table 1. The possibility of flat-faults at times 4 and 9 for window size τ_l is ruled out as we have argued in the proof of Proposition 3.2. Possible flat-faults for window size τ_s at times 6 and 11 (not shown) need to be examined, but cannot occur since the references at times 3, 4, 5, and 6 have to be the same, as implied by the τ_l characterization.

Table 5 : $(S(\tau_l),S(\tau_s)) \in \mathbf{WSS}_{nn}(\tau_l,\tau_s)$, $(S(\tau_l),S(\tau_s)) \notin \mathbf{WSS}_{ff}(\tau_l,\tau_s)$										
time	1	2	3	4	5	6	7	8	9	10
$R \in \mathbf{R}_{nn}(\tau_l,\tau_s)$	a	b	c	c	c	c	b	a	a	a
$S(\tau_l=4)$	1	2	3	3	2	1	2	3	3	2
$S(\tau_l=3)$	1	2	3	2	1	1	2	3	2	1

The third string pair (in Table 6) shows that there is no reference string in $\mathbf{R}_{nn}(\tau_l,\tau_s)$ which can possess that *wss* string pair. The non-existence of a flat-fault free reference string can be demonstrated by an exhaustive search of all possible strings that can be generated. The example shown assumes that the page set contains four pages *a*, *b*, *c*, and *d*. Flat-faults are indicated by asterisks.

Table 6 : $(S(\tau_l),S(\tau_s)) \in \mathbf{WSS}_{ff}(\tau_l,\tau_s)$, $(S(\tau_l),S(\tau_s)) \notin \mathbf{WSS}_{nn}(\tau_l,\tau_s)$								
time	1	2	3	4	5	6	7	8
$R \in \mathbf{R}_{ff}(\tau_l,\tau_s)$	a	b	a	c	c	a	d	b
$S(\tau_l=6)$	1	2	2	3	3	3	4	4
$S(\tau_s=3)$	1	2	2	3	2	2	3	3^*

The exhaustive search of all possible string generations is presented in Table 7, where X's and their superscripts, l for τ_l and s for τ_s, indicate conflicts with the specification(s) of the corresponding *wss* string(s). The last X, for instance, is due to the simultaneous need of pages *a* and *b* in order to meet both *wss* string specifications.

Table 7 : All Possible String Generations								
time	1	2	3	4	5	6	7	8
I	a	b	b	X^s	-	-	-	-
II	a	b	a	c	a	a	X^s	-
III	a	b	a	c	c	a	d	$X^{s,l}$
IV	a	b	a	c	a	c	d	$X^{s,l}$

Q.E.D.

Proposition 3.7

The relationships among **BPCTT**(τ_l,τ_s), **BPCTTC**(τ_l,τ_s), **WSS**$_{ff}(\tau_l,\tau_s)$, and **WSS**$_{nn}(\tau_l,\tau_s)$ are

$$\mathbf{BPCTT}(\tau_l,\tau_s)\supset\mathbf{WSS}_{nn}(\tau_l,\tau_s)\cup\mathbf{WSS}_{ff}(\tau_l,\tau_s)\supset\mathbf{WSS}_{nn}(\tau_l,\tau_s)=\mathbf{BPCTTC}(\tau_l,\tau_s)$$

where the inclusion relationships are proper.

Proof : The first inclusion relationship can be easily verified. Every element in the set **WSS**$_{nn}(\tau_l,\tau_s)\cup$**WSS**$_{ff}(\tau_l,\tau_s)$ corresponds to some reference string. The characterization with either τ_s or τ_l must by Proposition 3.1 be individually a *bpc* string with $d_t < s_t$ for all t's. Furthermore, $S(\tau_l)\geq S(\tau_s)$ is implied by the inclusion property of the working set and by $\tau_l>\tau_s$. Therefore, the first inclusion relationship holds. The properness of the inclusion relationship can be shown by the example in Table 8. The string pair in the table can be verified easily to be in the set **BPCTTC**(τ_l,τ_s). The X in Table 8 indicates the abnormal termination of the string generation process (i.e., no reference string can be generated with the given *wss* string pair).

Table 8 : An Example of **BPCTTC**(τ_l,τ_s) String Pair						
time	1	2	3	4	5	6
$S(\tau_l=4)$	1	1	1	2	2	3
$S(\tau_s=3)$	1	1	1	2	2	1
R	a	a	a	b	b	X

The second proper inclusion relationship is a direct consequence of Proposition 3.6 and the definition of set union. The equality part is nothing but the restatement of Proposition 3.5 within our framework.

Q.E.D.

It is clear now that there must exist a set of properties less restrictive than those used in Proposition 3.5 which the string pairs in the set **BPCTT**(τ_l,τ_s) have to satisfy in order to be members of the union of the sets **WSS**$_{ff}(\tau_l,\tau_s)$ and **WSS**$_{nn}(\tau_l,\tau_s)$. It can be shown that these necessary and sufficient conditions do exist with the explicit introduction of flat-fault characterizations [Ferr82a], i.e., of two boolean-valued flat-fault strings indicating whether there is a flat-fault at each time with respect to the given pair of working set window sizes. Although it is difficult to implicitly incorporate the flat-fault characterizations into a set of sufficient and necessary conditions which *wss* string pairs in set **BPCTT**(τ_l,τ_s) must satisfy in order to be members of the union of the sets **WSS**$_{ff}(\tau_l,\tau_s)$ and **WSS**$_{nn}(\tau_l,\tau_s)$, the existence of these conditions is assured by the hierarchical structure among the related sets of string pairs as shown in Proposition 3.7.

4. Probability and Limiting Behavior of Flat-faults

Let us assume that page set $P=\{1,2,\cdots,n\}$ is referenced in string R where $R=r_1 r_2 r_3 \cdots r_t \cdots$. Four properties are presented here which are related to the probability of a flat-fault and the limiting behavior of flat-faults. Each reference in the string to the set S is statistically independent of the others unless stated otherwise.

Proposition 4.1

If each page in the set P is referenced with equal probability, then the probability P_{unif} of a flat-fault at time t, where $t>\tau$, is given by $(1-\frac{1}{n})(1-\frac{2}{n})^{\tau-1}$.

Proof : The probability that a flat-fault occurs at time t, where $t>\tau$, is the same as the probability that $r_t=x$, $r_{t-\tau}=y$, $x\neq y$, and $x,y\notin W(t-1,\tau-1)$. This latter event has the following probability :

$$P_{unif} = \frac{2\times C(n,2)\times(n-2)^{\tau-1}}{n^{\tau+1}} = \frac{n(n-1)(n-2)^{\tau-1}}{n^{\tau+1}},$$

that can be easily reduced to $P_{unif} = (1-\frac{1}{n})(1-\frac{2}{n})^{\tau-1}$.

Q.E.D.

The result of Proposition 4.1 can be generalized to the non-uniform case without too much difficulty, as shown in Proposition 4.2.

Proposition 4.2

If page i in the set P is referenced with probability p_i, then the probability P_{skew} of a flat-fault at time t, where $t>\tau$, is given by $\sum_{i=1}^{n}\sum_{j=i+1}^{n} 2p_i p_j(1-p_i-p_j)^{\tau-1}$.

Proof : The probability that a flat-fault occurs at time t, where $t>\tau$, is the same as the probability that $r_t=x$, $r_{t-\tau}=y$, $x\neq y$, and $x,y\notin W(t-1,\tau-1)$. This latter event has the following probability :

$$P_{skew} = \sum_{i=1}^{n}\sum_{j=1, j\neq i}^{n} p_i p_j \prod_{k=1}^{\tau-1}(1-p_i-p_j) = \sum_{i=1}^{n}\sum_{j=1, j\neq i}^{n} p_i p_j(1-p_i-p_j)^{\tau-1}$$

$$= \sum_{i=1}^{n}\sum_{j=i+1}^{n} 2p_i p_j(1-p_i-p_j)^{\tau-1}$$

Q.E.D.

It is not difficult to verify that, when $p_i=\frac{1}{n}$ for all i, $P_{skew}=P_{unif}$.

The impact of the uniform assumption on the probability of a flat-fault is an interesting subject to investigate; the following proposition gives an asymptotic result for this case.

Proposition 4.3

As τ approaches infinity, $P_{unif} \le P_{skew}$.

Proof : For simplicity, let $m=\tau-1$ and $del(n,m)=\frac{n^{m+1}}{(n-2)^m}(P_{unif}-P_{skew})$. We shall show that as $m\rightarrow\infty$, $del(n,m)\le 0$ is true. With some algebra, we can obtain the following :

$$del(n,m)=n-1-\sum_{i=1}^{n}\sum_{j=i+1}^{n}2p_ip_jn\left(\frac{n-n(p_i+p_j)}{n-2}\right)^m$$

Without loss of generality, assume that, for some k, $p_k<\frac{1}{n}$. Such k must exist unless we have the uninteresting case for which all p_k's are equal. This case yields $P_{skew}=P_{unif}$ for all τ. All p_k's cannot be greater than $\frac{1}{n}$, otherwise their sum would be greater than 1. We shall prove the proposition by considering the two following cases :

(1). There is an $l\neq k$ for which $p_l<\frac{1}{n}$. It is easy to see that, since $p_k+p_l<\frac{2}{n}$ or $n(p_k+p_l)<2$, the $\left(\frac{n-n(p_k+p_l)}{n-2}\right)^m$ term will approach infinity as $m\to\infty$. This shows that $del(n,m)<0$ as $m\to\infty$.

(2). All p_i's (with $i\neq k$) are greater than or equal to $\frac{1}{n}$. There must be at least one $p_l(l\neq k)$ that is strictly greater than $\frac{1}{n}$. If we have more than one such p_l, each one of them must satisfy $p_l-\frac{1}{n}<\frac{1}{n}-p_k$, otherwise the sum of all the p_i's would be greater than 1. This is equivalent to $n(p_l+p_k)<2$, and we can repeat the above argument. A slightly more complicated case is that when $p_k<\frac{1}{n}$, $p_l>\frac{1}{n}$, and all other p_i's are equal to $\frac{1}{n}$. For simplicity, let $p_k=\frac{1-\delta}{n}$ and $p_l=\frac{1+\delta}{n}$ where $1\geq\delta>0$. Thus, the expression of $del(n,m)$ can be expanded as follows :

$$del(n,m) = n-1-((n-2)^2-(n-2))\frac{1}{n}-2\sum_{j=1}^{n-2}\frac{1+\delta}{n}\left(\frac{n-2-\delta}{n-2}\right)^m$$

$$-2\sum_{j=1}^{n-2}\frac{1-\delta}{n}\left(\frac{n-2+\delta}{n-2}\right)^m \quad -2n\left(\frac{1-\delta^2}{n^2}\right)$$

$$=\frac{4n-6}{n}-2(n-2)\frac{1+\delta}{n}\left(1-\frac{\delta}{n-2}\right)^m-2(n-2)\frac{1-\delta}{n}\left(1+\frac{\delta}{n-2}\right)^m-2\left(\frac{1-\delta^2}{n}\right)$$

In this expansion, it is easy to see that the term $\left(1+\frac{\delta}{n-2}\right)^m$ will approach infinity as $m\to\infty$. Therefore, $del(n,m)<0$ is true also in this case.

Q.E.D.

The last proposition establishes an upper bound for the limiting flat-fault rate of a reference string in which we do *not* assume that references are independent.

Proposition 4.4

The limiting flat-fault rate has an upper bound $\frac{n}{\tau+1}$ for $\tau\geq n-1$ as $t\to\infty$.

Proof : Every time t a flat-fault occurs, the page referenced at time $t-\tau$ drops out of the current working set. This particular page cannot be referenced in the interval $[t-\tau+1,t]$ of length τ. Therefore, every page in the set S can contribute to at most one flat-fault every $\tau+1$ references. If we let $|R|$ be the length of the reference string and UB an upper bound of the limiting flat-fault rate, then we have the following inequality :

$$\frac{n\left\lfloor\frac{|R|}{\tau+1}\right\rfloor}{|R|} \le UB \le \frac{n\left\lceil\frac{|R|}{\tau+1}\right\rceil}{|R|}$$

As we let $t \to \infty$ or equivalently $|R| \to \infty$, $UB = \frac{n}{\tau+1}$.

Q.E.D.

When $\tau = n-1$, the upper bound is tight, as shown by the following string R, where, except for the first few references, every reference generates a flat-fault :

$$123 \cdots n\,123 \cdots n\,123 \cdots n\,123 \cdots$$

This string R touches n pages which are referenced cyclically from 1 to n. With window size $\tau = n-1$, there is a flat-fault for every $t \ge n$.

5. Conclusions

The working set size strings obtained with one and two working set window sizes have been studied, with particular attention to the problem of flat-faults. For a single τ characterization, a simple set of necessary and sufficient conditions has been given for an integer string to generate a reference string [Ferr81a]. Working set size strings derived from a real trace can be used to generate flat-fault free reference strings even though the original real trace may contain flat-faults.

For a double τ characterization, a set of necessary and sufficient conditions has been derived for a pair of integer strings to correspond to a flat-fault free reference string [Ferr82a]. This set of conditions is only sufficient for the generation of a general class of reference strings with or without flat-faults. It is by no means necessary, as shown in Proposition 3.6. Therefore, a pair of working set size strings derived from a real trace needs to be checked against this set of conditions if it is to be used to generate a flat-fault free reference string. In other words, this set of conditions can be used to check whether there is any flat-fault in a given pair of *wss* strings extracted from a real trace. Although a set of necessary and sufficient conditions has been shown to exist for the general class of string generation algorithms, such conditions have not been formulated yet. Note that, when *wss* characterizations are accompanied by flat-fault characterizations, necessary and sufficient conditions have already been found [Ferr82a]. It is for the case in which flat-fault characterizations are not given that only sufficient conditions are known.

The probability of a flat-fault under the independent reference assumption has been obtained. In particular, we have shown that, when the working set window size is relatively large in comparison with the cardinality of the page set, this probability is bounded from below by the probability derived with the uniform page reference assumption. The probability for the latter case is easy to calculate and very small in practice. Finally, an upper bound for the limiting flat-fault rate has been derived.

Acknowledgements

The author is grateful to Professor Ferrari for his guidance on the subject of this paper and his ever-patient and careful proofreadings. The current version of proof for Proposition 3.2 has been improved as a result of Bob Hagmann's comments.

References

[Dutt81a]
C. R. Dutt, "Dynamic Characterization and Reproduction of Program Memory Consumption with Working-Set-Based Generative Model," Master's Report, Computer Science Division, University of California at Berkeley, August 1981.

[Fall81a]
N. Faller, "Design and Analysis of Program Models for the Reproduction of Working-Set Characteristics," Ph.D. dissertation, Computer Science Division, University of California at Berkeley, June 1981.

[Ferr81a]
D. Ferrari, "Characterization and Reproduction of the Referencing Dynamics of Programs," Performance '81, F. Kylstra, Ed., North-Holland, Amsterdam, November 1981, pp. 363-372.

[Ferr81b]
D. Ferrari, "A Generative Model of Working Set Dynamics," Proc. of the Sigmetrics Conference on Measurement and Modeling on Computer Systems, 1981, pp. 52-57.

[Ferr82a]
D. Ferrari, "Working Set Size Strings," PROGRES Report No. 82.10, Computer Science Division, University of California at Berkeley, December 1982.

[Lee82a]
T. P. Lee, "Experimental Design of a Generative Model Based on Working Set Size Characterizations," Master's Report, Computer Science Division, University of California at Berkeley, June 1982.

[Spir77a]
J. R. Spirn, "Program Behavior : Models and Measurements," Elsevier, New York, 1977.