ISSN: 2278 – 7798
International Journal of Science, Engineering and Technology Research (IJSETR)
Volume 2, Issue 7, July 2013

DISCOVERING REASONS FOR BELADY ANOMALY IN FIFO

YASHASVINI SHARMA

Abstract— **In computer storage, Belady's anomaly proves that it is possible to have more page faults when increasing the number of page frames while using the First in First out (FIFO) page replacement algorithm. László Bélády demonstrated this in 1969. Usually, on increasing the number of frames allocated to a process' virtual memory, the process execution is faster, because fewer page faults occur.**

Sometimes, the reverse happens, i.e., the execution time increases even when more frames are allocated to the process. This is Belady's Anomaly. This is true for certain page reference patterns.
The present work is an approach to find why Belady's anomaly occurs and under what condition it successfully appears. This means suppose we are provided with a memory storage of lesser and other with greater number of page frames then just by viewing we can say whether Belady's anomaly can occur in memory location having greater number of page frames or not.

Index Terms— **Belady's anomaly, virtual memory, demand paging, page replacement, FIFO**

I. Introduction

In common computer memory management; information is loaded in specific sized chunks. Each chunk is referred to as a page. The central processor can only load a limited number of pages at a time. It requires a frame for each page it can load. A page fault occurs when a page is not found, and might need to be loaded from disk into memory. When a page fault occurs and all frames are in use, one must be cleared to make room for the new page. A simple algorithm is FIFO. Whichever page has been in the frames the longest is the one that is cleared. Until Bélády's anomaly was demonstrated, it was believed that an increase in the number of page frames would always provide the same number or fewer page faults.

In Operating Systems (OS) courses, instructors often like to assign software projects that relate to course objectives. One topic with special appeal to students is Belady's anomaly. For a virtual memory system with demand paging, the page fault rate of a process varies with the number of memory frames allocated to the process. When in increase in the number of allocated, frames leads to an increase in the number of page faults (a bump).

We know that Belady's anomaly occurs in FIFO, but when it occurs is the important question. Actually for Belady's anomaly the pattern of arriving pages are quite same, and due to this anomaly occurs.

Here is an example of Belady's anomaly.

Page Requests	**3 2 1 0 3 2 4 3 2 1 0 4**
Newest Page	3 2 1 0 3 2 4 4 4 1 0 0
	3 2 1 0 3 2 2 2 4 1 1
Oldest Page	3 2 1 0 3 3 3 2 4 4
Page Requests	3 2 1 0 3 2 4 3 2 1 0 4
Newest Page	3 2 1 0 0 0 4 3 2 1 0 4
	3 2 1 1 1 0 4 3 2 1 0
	3 2 2 2 1 0 4 3 2 1
Oldest Page	3 3 3 2 1 0 4 3 2

An example of Bélády's ar
Using three page frames, !
faults occur. Increasing to
page frames causes 10 pa
faults to occur. Page faults
in red.

II. LITERATURE REVIEW

2.1 Methodology

This research study used computer simulation to examine conditions that affect how often Belady's anomaly will occur. We focused on four operating system conditions that affect the incidence of Belady's anomaly:
1. Page replacement algorithm.
2. Process size in pages.
3. Reference string length.
4. Number of memory frames allocated to the process.

2.2 Previous Works

Belady, L. A. (1966).[1] A study of replacement algorithms for a virtual-storage computer. IBM Systems Journal, Volume 5, Number 2.
Belady, L. A., et al. (1969).[2] An anomaly in space-time characteristics of certain programs running in a paging machine. CACM, Volume 12, Number 6.
Mattson, R. L., et al. (1970).[3] Evaluation techniques for storage hierarchies. IBM Systems Journal, Volume 9, Number 2.
A Probability Model for Belady's Anomaly[4] Kirby McMaster, Samuel Sambasivam, Nicole Anderson

III. THE PROPOSED METHOD

3.1 Assumptions

For finding when Belady's anomaly can occur, we have to take following assumptions
Before that let's took an overview how it occurs

Suppose in computer storage
1st memory storage having smaller number of page frame=x
2nd memory storage having greater number of page frame=t

And let's take y=t+1
Such that x<t and x<y

.

3.2 The Working Model explanation with an Example

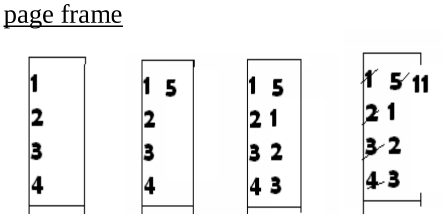
To understand why Belady's anomaly occurs, the technique used by memory storage having smaller number of page frame is

1). To preserves the pages earlier that will be used in future
In FIFO one of the best techniques to increase number of page fault is, to overwrite recent page with any of the new page and
then calling the overwritten page back.
This merely makes the page faults equal in memory storage having smaller and greater number of page frame
Example
Memory storage having smaller number of page frame is 4
Memory storage having greater number of page frame is 5
And all the frames are filled with required pages in both of the memory storage, of coarse the larger one will hold an extra page than
that of the memory storage having smaller number of page frame.

i) We know that any need of new page, that is not in either of the memory allocation will cause page fault in both types of memory storages.
ii) Memory storage having greater number of page frame has more number of pages.
iii) Firstly we need to delete a page from memory storage having greater number of page frame and calling back same page

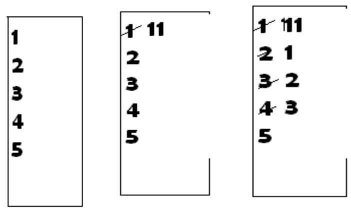
Pages are 1,2,3,4,5,1,2,3,11,1,2,3

<u>In memory storage having smaller number of page frame</u>



Number of Page faults
4 1 3 1
Total page fault=4+1+3+1=9

<u>In memory storage having greater number of page frame</u>



Number of Page faults

5 1 3

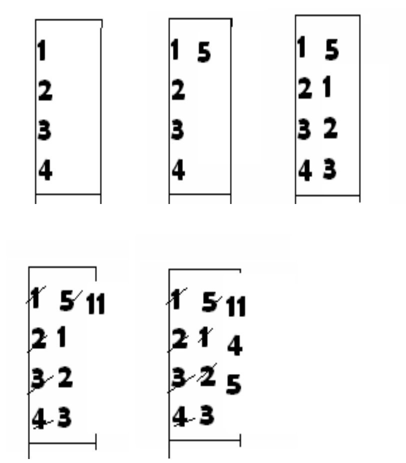
Total page fault=5+1+3=9

2). To maintain the preserve pages in memory storage having smaller number of page frame, and try to replace these pages in memory storage having greater number of page frame.

Extending above example

1,2,3,4,5,1,2,3,11,1,2,3,4,5,11

<u>In memory storage having smaller number of page frame</u>

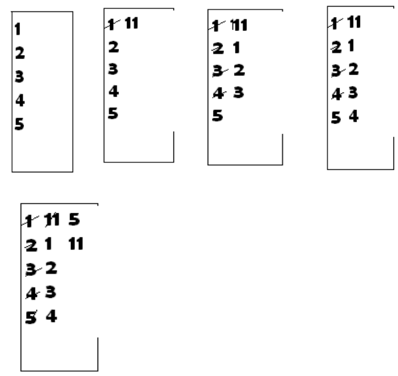


Number of Page faults

4 1 3 1 2

Total page fault=4+1+3+1+2=11

<u>In memory storage having greater number of page frame</u>



Page fault

5 1 3 1 2

Total page fault=5+1+3+1+2=12

So here conclusion withdrawn that we can maintain our preserved page 11 in memory storage having smaller number of page frame

The more space in memory storage having smaller number of page frame we have more chances of preserving those pages that are there in memory storage having greater number of page frame.

Steps that will make concept clear of above example

1) We save 4 pages namely 1, 2, 3, 4 in memory storage having smaller number of page frame. When page number 5 will come because of FIFO it will remove page number 1 and takes its position.

1, 2,3,4,5,1,2,3

2) Then the 4th place which is holding 5 is replaced by a new page 11 that is not in memory storage having greater number of page frame

1,2,3,4,5,1,2,3,11

3) In memory storage having greater number of page frame, 11 overwrites 1,1overwrites 2, 2 overwrites 3 and 3 overwrites 4.

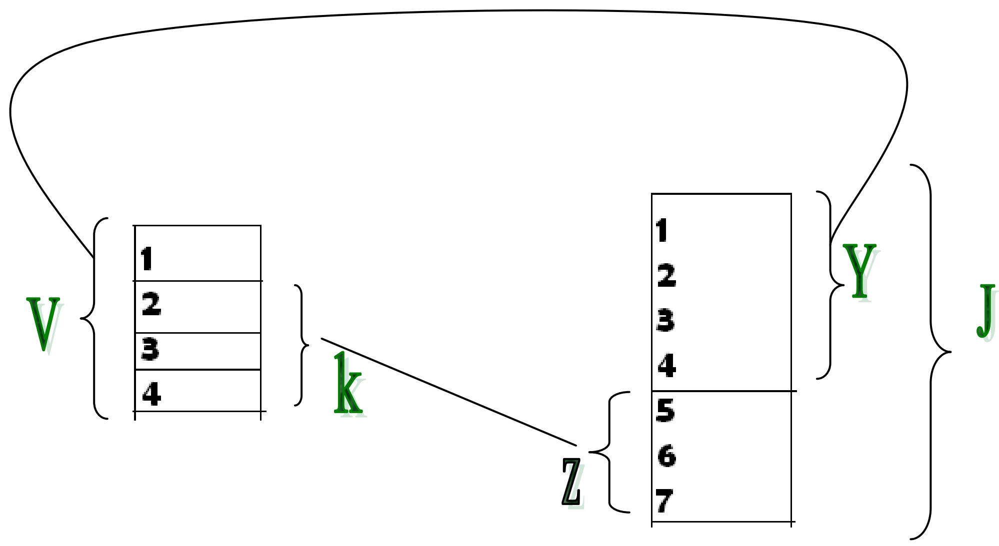
1,2,3,4,5,1,2,3,11,1,2,3

Till here page faults are same

4) Here biggest thing is number of turn taken to reach 11 in memory storage having greater number of page frame, which is 2 as 11, is the nearest page in memory storage having greater number of page frame, which is also in memory storage having smaller number of page frame. But in memory storage having smaller number of page frame, to overwrite 11 we still are having 3 page frames to go, which is greater than number of turn taken to reach 11 in

memory storage having greater number of page frame, which is 2.

frame then only we can make Belady's anomaly else not.



memory location having lesser and greater number of page frames

So if we call page frames as
1,2,3,4,5,1,2,3,11,1,2,3,4,5,11
The upcoming 5 replaces 11 in memory storage having greater number of page frame which is still in memory storage having smaller number of page frame, so if we call 11, the memory storage having greater number of page frame will issue a page fault, but memory storage having smaller number of page frame will not and thus Belady's anomaly occurs.

But if we try it for
Memory storage having smaller number of page frame is 4

Memory storage having greater number of page frame is 7
There is no way of developing a string that will show Belady's anomaly
It can never happen as we cannot maintain our preserved page 11 in memory storage having smaller number of page frame because in FIFO we actually move in a circular path if we can traverse the circular path in memory storage having greater number of page frame and replace that page in this movement that is still in memory storage having smaller number of page

So from here we got a clue, from which the above formula is developed
Let
Memory storage having smaller number of page frame is V
Memory storage having greater number of page frame is J
J=Y+Z
Where Y, Z, V, k all are number of page frames
Y=V
And k=V-1
If k=<Z, then no Belady's anomaly can ever occur

3.3 Diagrammatical Overview

This is diagrammatical solution of Belady's anomaly and it is same as that of written in method part, by this one can check whether Belady's anomaly will occur or not.

This can be explained as let 7= y + z, where y and z are number of page faults.Here it will make no sense to take y>4 which is memory storage having smaller number of page frame because if we extend it will replace that particular, is memory storage having smaller number of page frame that we want to

preserve and this process will continues if we increase y more than memory storage having smaller number of page frame .This y part is for doing first point explained in this topic.
So largest value of y we can take is 4 and z is 7-y, as z increases the possibility of completing the circular path decreases; here memory storage having smaller number of page frame is 4, which means if we want Belady's anomaly to occur then at least we have to preserve one page, this proves that we can overwrite rest 3 page frame and if value of z>=3 then we cannot complete our circular path in memory storage having greater number of page frame is 7, so Belady's anomaly cannot occur

3.4 The Mathematical Model

The method for finding when Belady's anomaly can occur is quite simple
Suppose in computer storage
Number of page frame in 1^{st} memory storage having smaller number of page frame=x
2^{nd} memory storage having greater number of page frame=t
And let's take y=t+1
Such that $x<t$ and $x<y$
If y % x = 0 (after dividing y by x and we get remainder as 0)
Then for all p page frame such that p>=y
No Belady's anomaly can ever occur.

Explanation by an example
Let
x=4
t=7
Then y=7+1=8
$4<7$ and $4<8$
8%4=0(as we get remainder as 0)
So no Belady's anomaly can ever occur for
x=4
p>=7

IV. Conclusions

4.1 CONCLSION

By this we can conclude that when the Belady's anomaly follow the above given pattern then if the
1^{st} memory storage having larger number of page frame whose page frame is completely divisible by
The page frame in 2^{nd} memory storage having smaller number of page frame then no Belady's anomaly can ever occur.

References

[1] A Probability Model for Belady's Anomaly Kirby McMaster kmcmaster@smcm.eduCS Dept., St. Mary's College of MD St. Mary's City, MD 20686 Samuel Sambasivam ssambasivam@apu.edu
CS Dept., Azusa Pacific University Azusa, CA 91702 Nicole Anderson nanderson@winona.edu CS Dept., Winona State University Winona, MN 55987
[2] Belady, L. A. (1966). A study of replacement algorithms for a virtual-storage computer. IBM Systems Journal, 5(2).
[3]Belady, L. A., Nelson, R. A., & Shedler, G. S. 1969. An anomaly in space-time characteristics of certain
programs running in a paging machine. Communications of the ACM, 12(6), 349 - 353.
[4]Deitel, P., & Choffnes, D. (2004). Operating systems. Prentice Hall.
[5]Dhamdhere, D. (2008). Operating systems. McGraw-Hill.
[6]Franaszek , P. A., & Wagner, T. J. (1974). Some distribution-free aspects of paging algorithm performance. Journal of the ACM, 21(1).
[7]Fine, G. H., Jackson, C. W., & McIsaac, P. V. (1966). Dynamic program behavior under paging. Proceedings of the 1966 21st National ACM Annual Conference, pp. 223 – 228.

Author

Yashasvini Sharma, student of BE computer science branch, seventh semester, pursuing from Gyan Ganga Institute of Technology and Sciences, Jabalpur, Madhya Pradesh, India.