David Lybeck
CS 475
Professor Chiu
2 May 2024

Belady's anomaly is a counterintuitive-seeming phenomenon observed in paging systems. It is where decreasing the amount of memory available to a process can sometimes lead to improved performance, (or increasing memory can lead to decreased performance) measured by a change in the number of page faults or execution time. This anomaly challenges the general idea that providing more memory should result in better performance in a system. The anomaly was first discovered and analyzed by Belady, Nelson, and Shedler in the Thomas Watson Research Center at IBM in 1969[1] and has been further researched since then.

To understand Belady's anomaly well, it is a good idea to have a basic understanding of paging and replacement algorithms and how they are used inside of virtual memory systems. Paging is a memory management technique that is used in operating systems to allow processes to access more memory than is physically available in the machine (virtual memory). The process's address space is divided into fixed-size pages, and then these pages are loaded into physical memory (also divided into page frames) as they are needed during execution. When a process attempts to access a page that is not currently in physical memory, a page fault occurs. The operating system must then load the required page into a free page frame all the way from disk, which can take a very long time in comparison to normal, potentially evicting an existing page from memory. The decision of which page to evict is governed by a replacement algorithm. According to the article “Belady’s Anomaly in Page Replacement Algorithms.” Two common replacement algorithms are FIFO (First in first out) and LRU (Least Recently Used), as well as an algorithm know as Optimal Page Replacement Algorithm[2]. LRU and Optimal Page Replacement Algorithm are examples of replacement algorithms that are stack based, so they avoids the problem of belady's anomaly[3].

Belady's anomaly arises due to the interplay between the page reference pattern of a process and the replacement algorithm used by the operating system. Consider two memory configurations, L (larger) and S (smaller), with the same page reference string and replacement algorithm (such as FIFO). Under certain conditions, it is possible for the smaller memory configuration S to experience fewer page faults than the larger configuration L, even though the intuitive expectation is that more memory should lead to better performance. Here are some examples demonstrating this phenomenon:

---

[1] 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
[2] “Belady’s Anomaly in Page Replacement Algorithms.”
[3] Kochar, Ankit. “Belady’s Anomaly.”

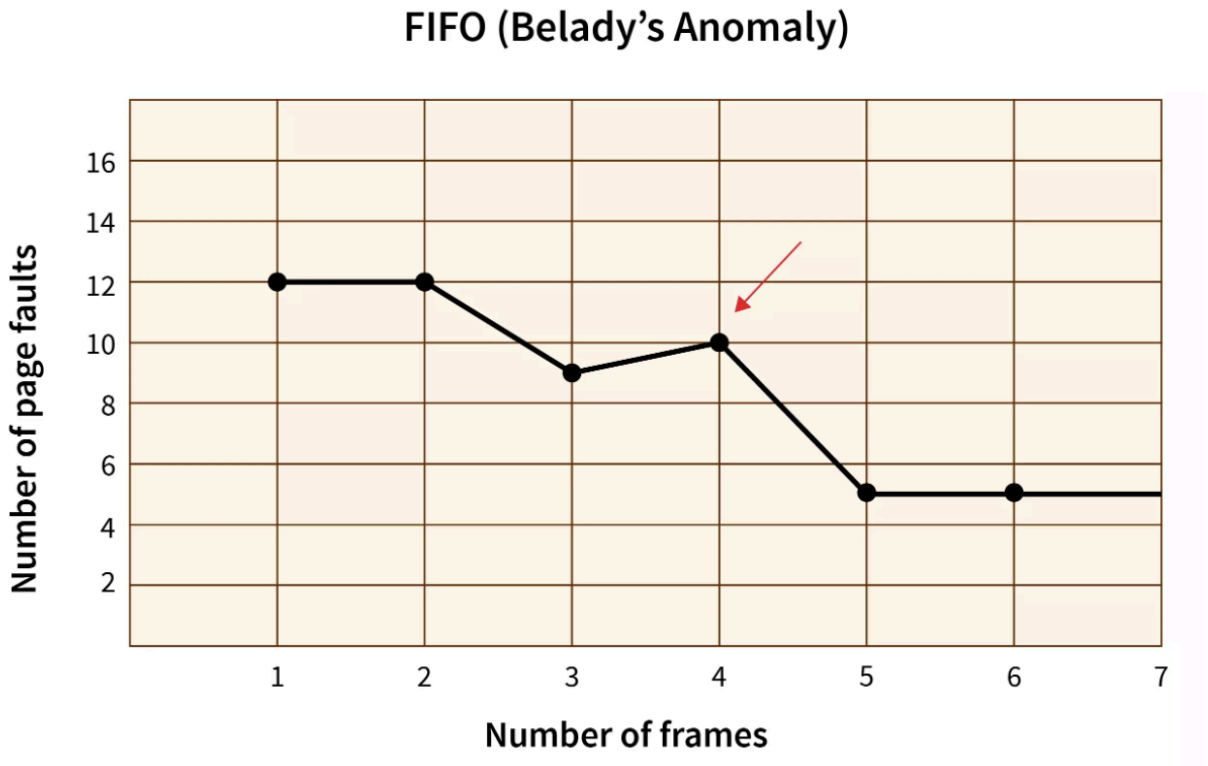

Image from [1] Aman. "Belady's Anomaly in Operating System." www.scaler.com/topics/beladys-anomaly/

Frames added in the pattern: 1, 2, 3, 4, 1, 2, 5, 1, 2, 3, 4, 5
PF = Page Fault

9 page Faults with 3 frames:

1	1	1	2	3	4	1	1	1	2	5	5
	2	2	3	4	1	2	2	2	5	3	3
		3	4	1	2	5	5	5	3	4	4
PF	PF	PF	PF	PF	PF	PF	X	X	PF	PF	X

10 page faults with 4 frames:

1	1	1	1	1	1	2	3	4	5	1	2
	2	2	2	2	2	3	4	5	1	2	3
		3	3	3	3	4	5	1	2	3	4
			4	4	4	5	1	2	3	4	5
PF	PF	PF	PF	X	X	PF	PF	PF	PF	PF	PF

Images from "Belady's Anomaly in Page Replacement Algorithms." www.geeksforgeeks.org/beladys-anomaly-in-page-replacement-algorithms/

There are many reasons that lead belady's anomaly to occur, but one of the key reasons is that "increasing the number of frames can lead to different page eviction patterns and a page that was previously not evicted may now get replaced, causing additional page faults." (upgrad)

Belady provided two production rules, the "cluster rule" and the "cyclic rule," to generate page reference strings that exhibit the anomaly under the FIFO replacement algorithm.[4] These rules demonstrate how specific patterns of page references can lead to the counterintuitive behavior observed in Belady's anomaly. Ultimately these come from the stack-based nature of First in First Out, meaning that in FIFO the programs that are removed first are the ones that have been there the longest, while in LRU or in Optimal Page Replacement Algorithm, they remove the processes that are used the least, keeping the more important and commonly used processes, avoiding Belady's Anomaly[5]. Along with this they also have a consistent replacement policy, meaning no matter the size of the memory, the same processes will get removed. And They utilize temporal locality, which means that if a process is recently used, it is likely to be used again in the future. The number of frames in memory does not affect the use of these types of algorithms[6].

Belady's anomaly has significant implications for the design and analysis of virtual memory systems and replacement algorithms. It highlights the importance of considering not only the memory size but also the specific access patterns of processes when evaluating system performance. The anomaly also challenges the traditional assumption that providing more memory always leads to better performance. While this assumption generally holds true, Belady's anomaly demonstrates that there can be exceptions, particularly when the memory access patterns of processes interact with the replacement algorithm in specific ways. Understanding and accounting for Belady's anomaly is crucial for accurate performance modeling and optimization of virtual memory systems. It has inspired further research into alternative replacement algorithms and memory management techniques that aim to mitigate or exploit the anomalous behavior observed in certain scenarios.

Belady's anomaly is an interesting and counterintuitive phenomenon in virtual memory management, along with many other areas such as legacy systems, caching and more[7]. It demonstrates that increasing the amount of memory available to a process can sometimes lead to decreased performance, challenging the intuitive expectation that more memory should always result in better performance. This anomaly arises due to both the page reference pattern of a process and the replacement algorithm used by the operating system. While the anomaly's surprising at first, it shows the importance of considering not just the memory size, but also the specific access patterns of processes when they're evaluating system performance. Understanding and accounting for Belady's anomaly is very important for accurate performance modeling and optimization of virtual memory systems, and if not properly accounted for, Belady's Anomaly can cause issues to system performance in general. Decreasing overall performance as well as decreasing response time for users[8]. Because of these problems, and

---

[4] Belady, L. A., Nelson, R. A., & Shedler, G. S. (1969). An anomaly in space-time characteristics

[5] Kochar, Ankit. “Belady’s Anomaly.”

[6] Aman. “Belady’s Anomaly in Operating System.”

[7] VATS, ROHAN. “Understanding Belady’s Anomaly: A Puzzling Memory Management Phenomenon.”

[8] Datta, Subham. “What Is the Belady’s Anomaly?”

the anomaly's odd cases, it has inspired further research into alternative memory management techniques. Overall, Belady's anomaly serves as a reminder that intuition alone is not always sufficient in the field of computer science, and that rigorous analysis and empirical evaluation are essential for understanding and optimizing complex systems.

Works Cited

Aman. “Belady’s Anomaly in Operating System.” Scaler Topics, Scaler Topics, 29 Aug. 2022, www.scaler.com/topics/beladys-anomaly/.

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. https://doi.org/10.1145/363011.363155

“Belady’s Anomaly in Page Replacement Algorithms.” GeeksforGeeks, 2 May 2023, www.geeksforgeeks.org/beladys-anomaly-in-page-replacement-algorithms/.

Datta, Subham. “What Is the Belady’s Anomaly?” Baeldung on Computer Science, 18 Mar. 2024, www.baeldung.com/cs/security-belady-anomaly.

Kochar, Ankit. “Belady’s Anomaly.” PrepBytes, 29 Dec. 2023, www.prepbytes.com/blog/operating-system/beladys-anomaly/.

VATS, ROHAN. “Understanding Belady’s Anomaly: A Puzzling Memory Management Phenomenon.” UpGrad, www.upgrad.com/tutorials/software-engineering/software-key-tutorial/beladys-anomaly/. Accessed 2 May 2024.