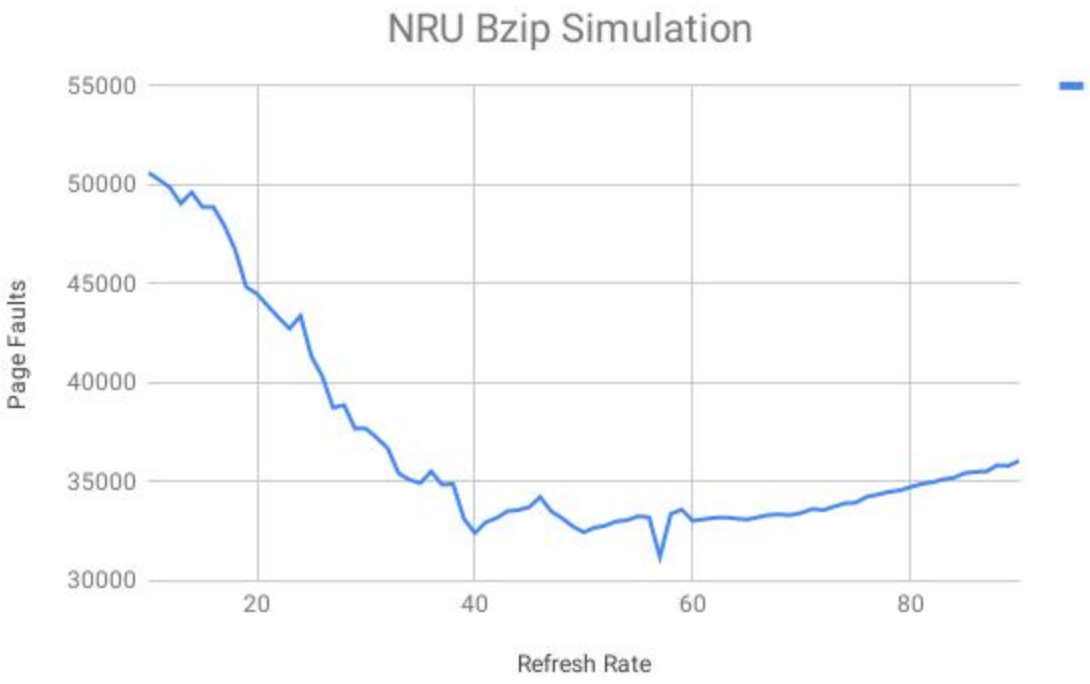
Michael Korst

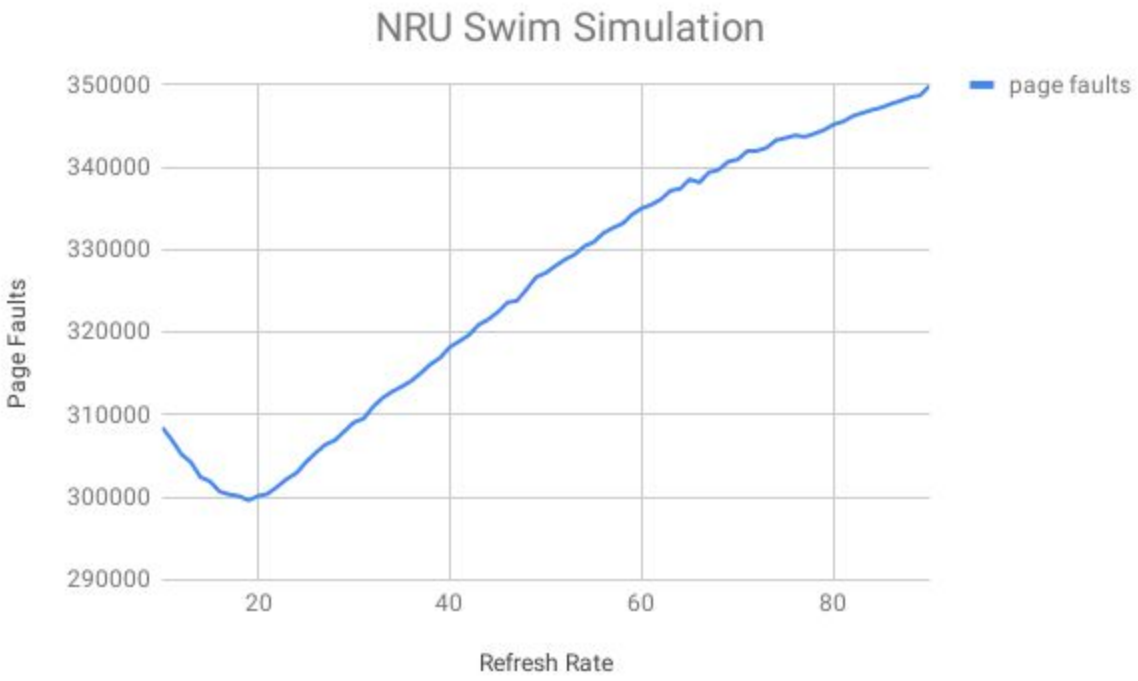
CS 1550

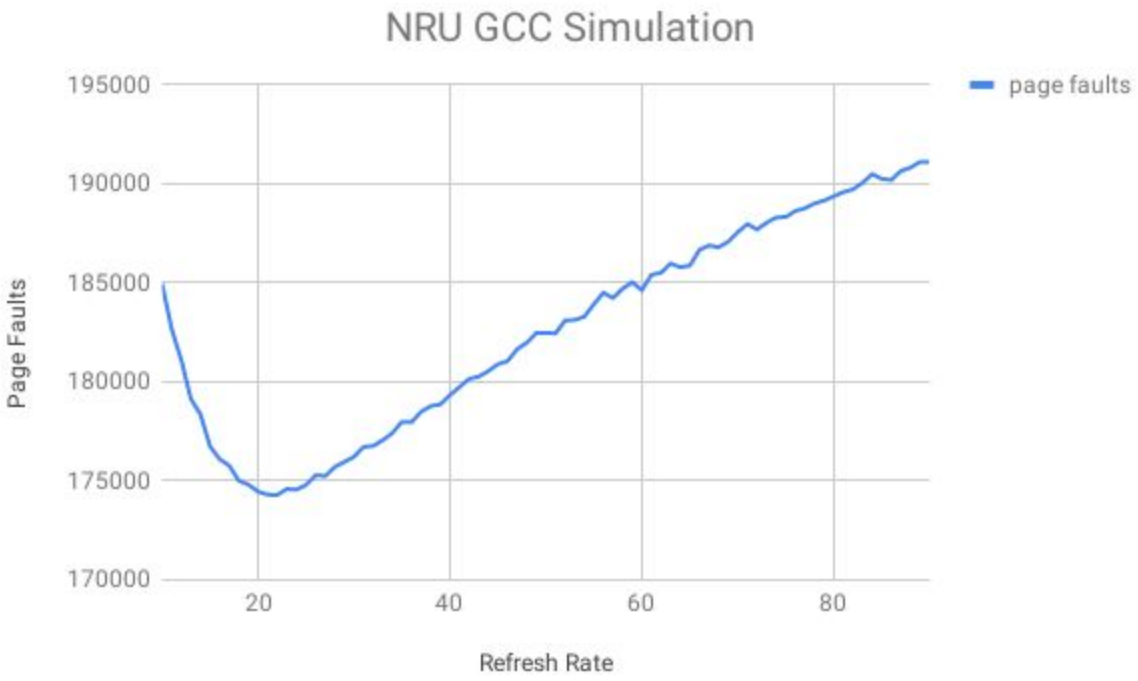
12 November 2018

Project 3 Write-Up

1) Determining optimal value for NRU refresh parameter, ranging refresh rate from 10 to 90 (for each of the 3 trace files):







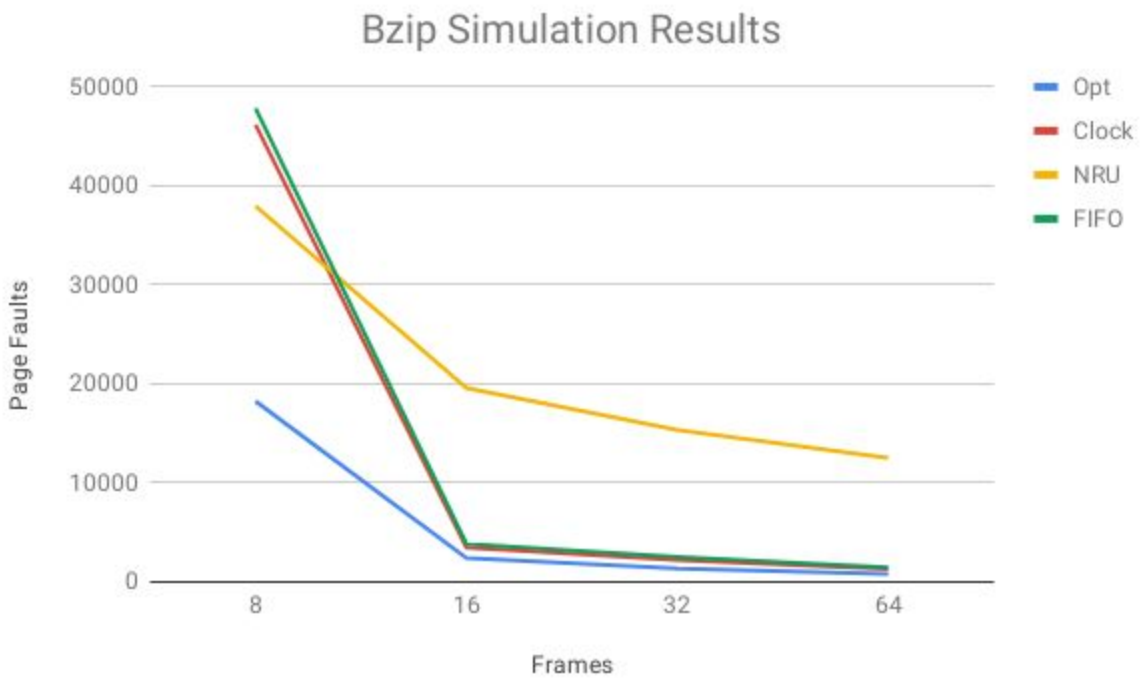
Based on iterated simulations done on the 3 trace files, the minimum of this range for page faults seems to be somewhere 20 for the last 2 and 45-50 for the 1st bzip

sim. Based on this, I think 30 would be a reasonable refresh rate to pick as “optimal”. (Note: 8 frames were used for all of these simulations).

2) Below is a comparison of page fault rates in all 4 algorithms for frame sizes of 8, 16, 32 and 64. For NRU the previously determined refresh of 30 is used.

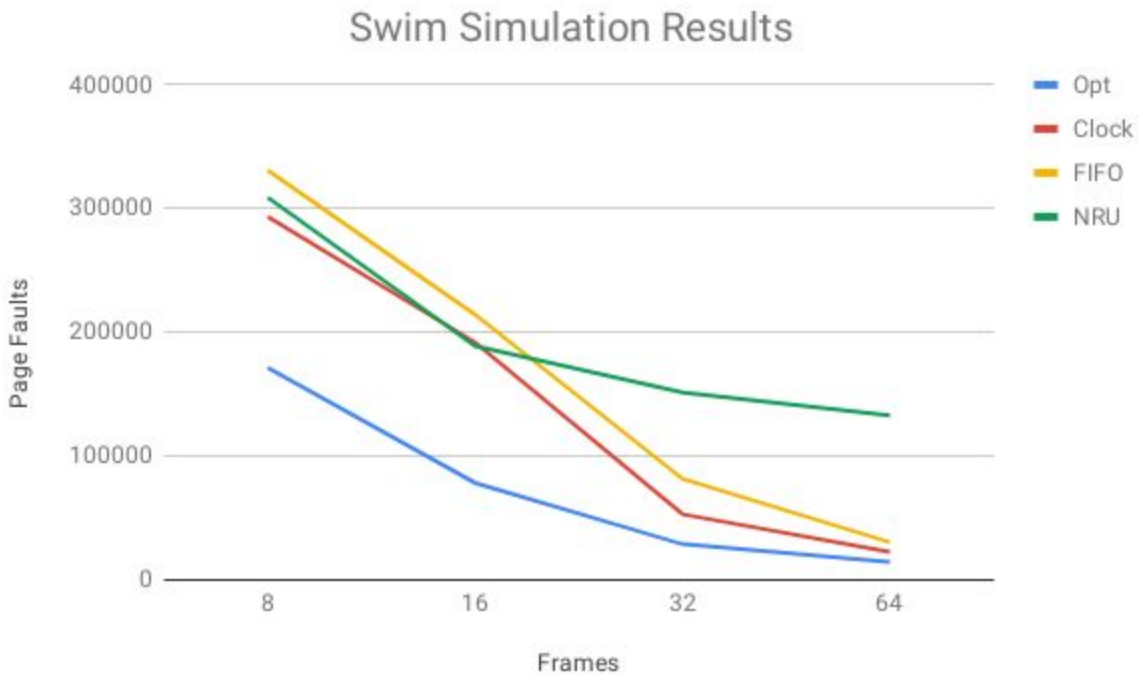
Bzip Sim Results:

Opt		Clock		FIFO		NRU	
Frames	Page Faults	Frames	Page Faults	Frames	Page Faults	Frames	Page Faults
8	18251	8	46164	8	47828	8	37953
16	2427	16	3468	16	3820	16	19589
32	1330	32	2203	32	2497	32	15340
64	821	64	1318	64	1467	64	12527



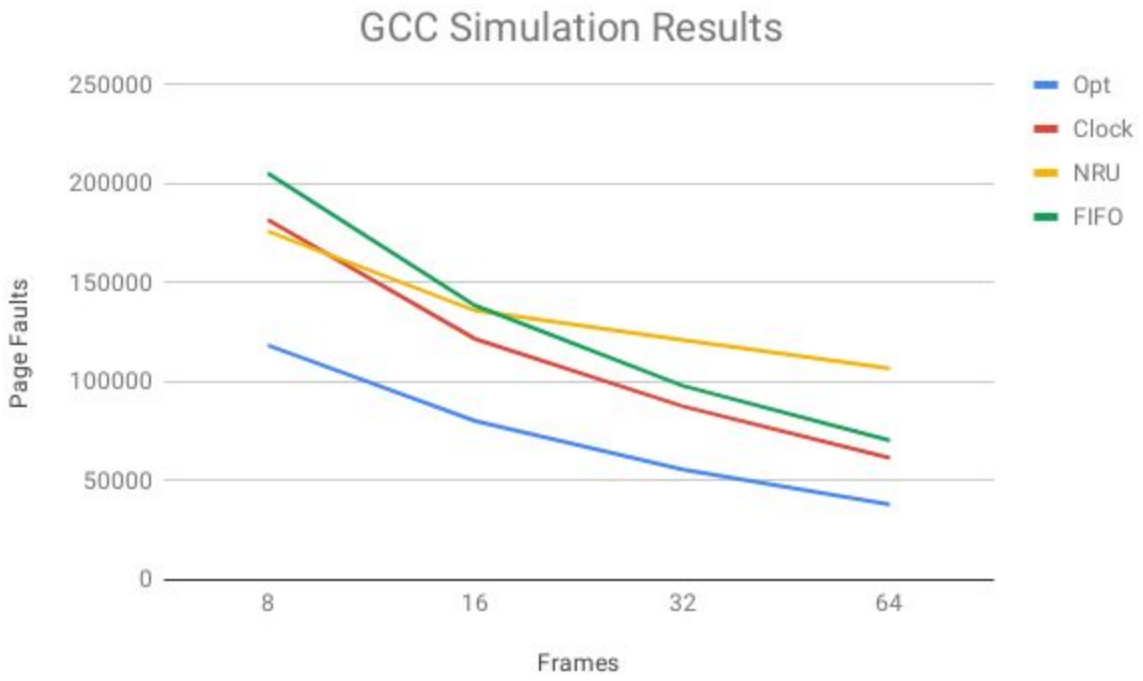
Swim Sim Results:

OPT		Clock		FIFO		NRU	
Frames	Page Faults	Frames	Page Faults	Frames	Page Faults	Frames	Page Faults
8	171244	8	293519	8	330893	8	309038
16	78312	16	191848	16	214295	16	188867
32	28826	32	53025	32	81638	32	151360
64	14289	64	22611	64	30422	64	132947



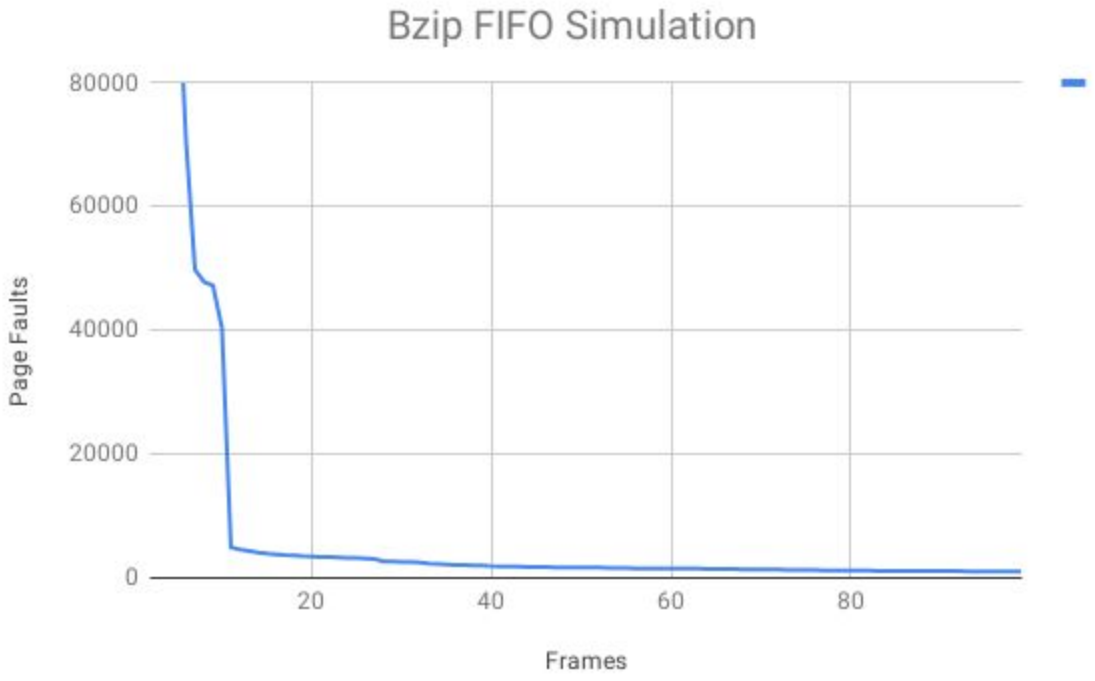
GCC Sim Results:

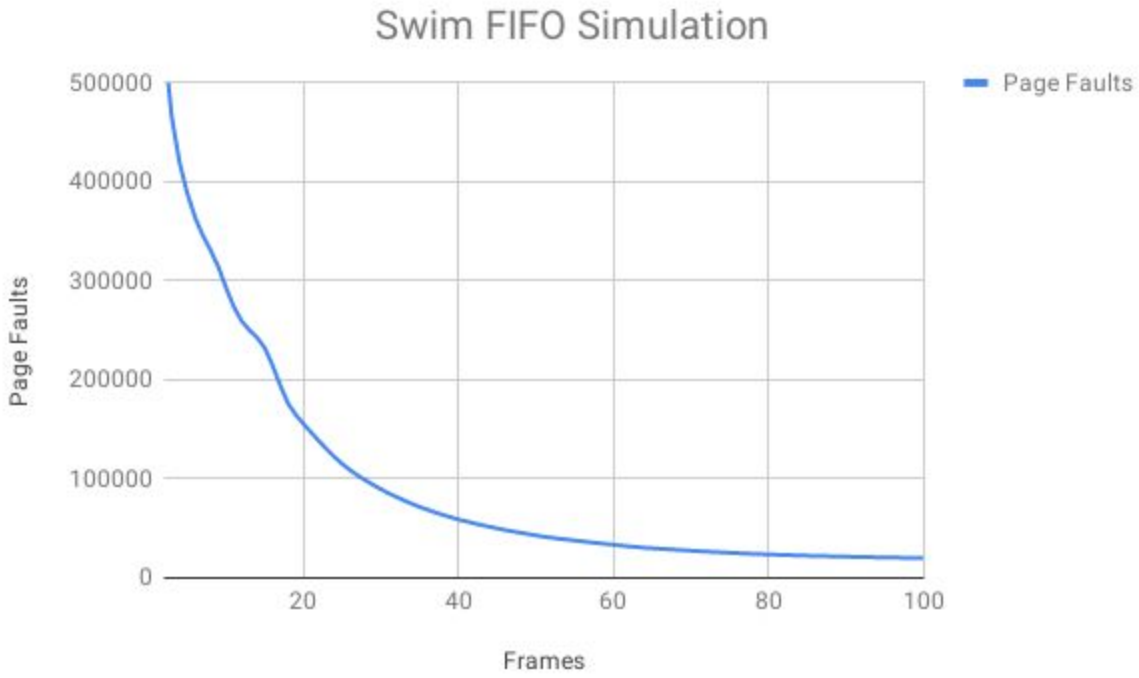
OPT		Clock		FIFO		NRU	
Frames	Page Faults	Frames	Page Faults	Frames	Page Faults	Frames	Page Faults
8	118480	8	181856	8	205368	8	176099
16	80307	16	121682	16	138539	16	135967
32	55802	32	87686	32	98067	32	121222
64	38050	64	61640	64	70315	64	106691

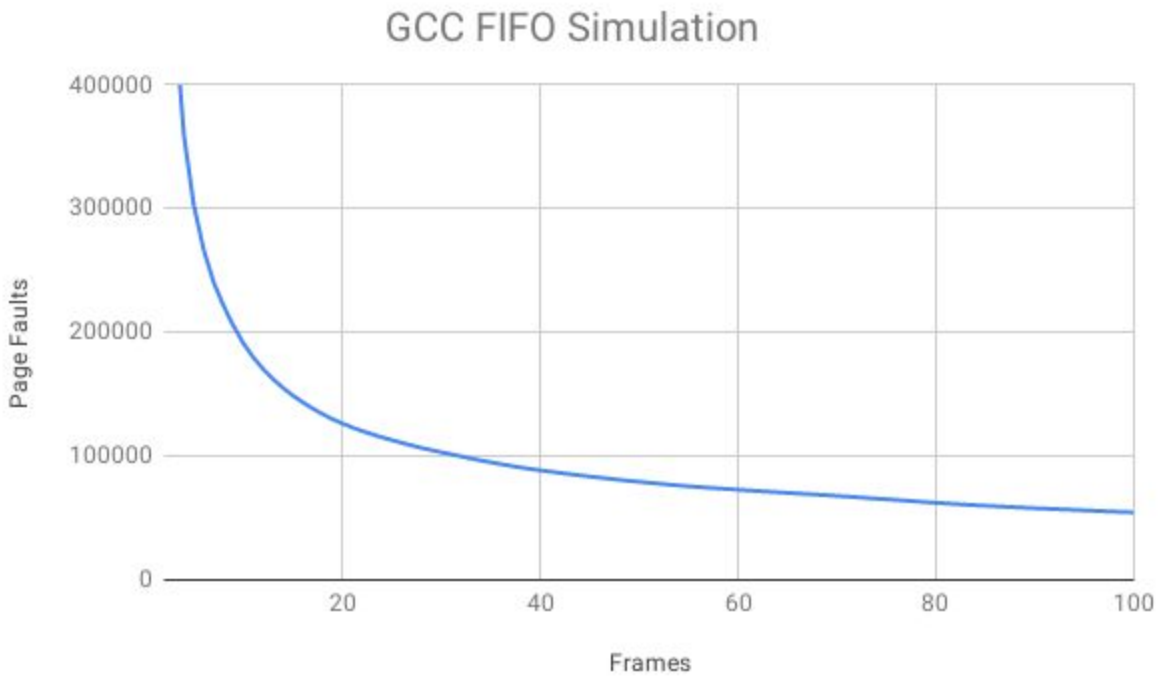


Conclusion: Based on the simulation numbers and charts, it appears that Clock is the closest algorithm to optimal. It certainly seems to largely follow the trends of Opt as it responds to an increase in frames. FIFO is, slightly unexpectedly, the second most optimal algorithm. It generally starts out the slowest but becomes more efficient with an increase in frames. NRU actually appears the least efficient. However, NRU is heavily dependent on an optimized refresh rate. In a real system, this could be determined by many different factors and varies. So, because of this, if optimized properly, NRU might be able to be the best algorithm. Additionally, even though FIFO seems alright on a smaller scale, it has no priority consideration and thus would likely not be a good choice for real OS design. So, based on these observations from the trace files, Clock is the algorithm with the least page faults, the closest pattern to Opt, and is thus the optimal algorithm for a real operating system.

3) Below is a chart showing the page fault rate for FIFO on each of the three traces, varying the frames from 2 to 100:







Based on the iterations shown above, I only found one instance of Belady's anomaly. This occurred as we increased from 91 to 92 frames on the Bzip simulation, with an increase in 4 page faults. Other than that one time, there is no instance of the anomaly where we would see the faults increasing with an increase in frames. The one anomaly I found is, indeed, an anomaly. However, it's still important to note the trends in FIFO with an increase in frames, as the page fault decrease seems to lessen over time. The algorithm doesn't appear to become much more efficient even with vast amounts of frames. This is why we generally avoid FIFO in Operating System design, because it's fairly naive and doesn't try to account for patterns in memory access..