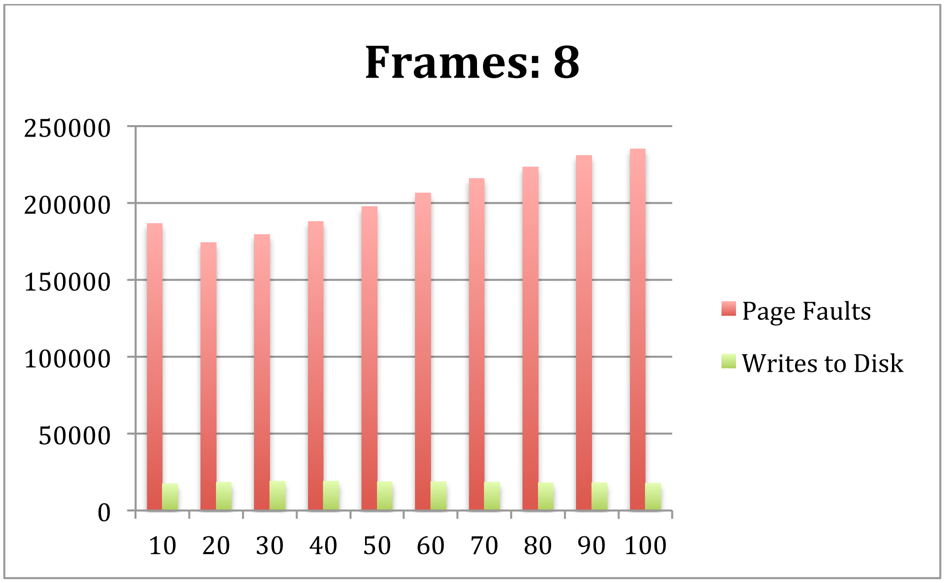
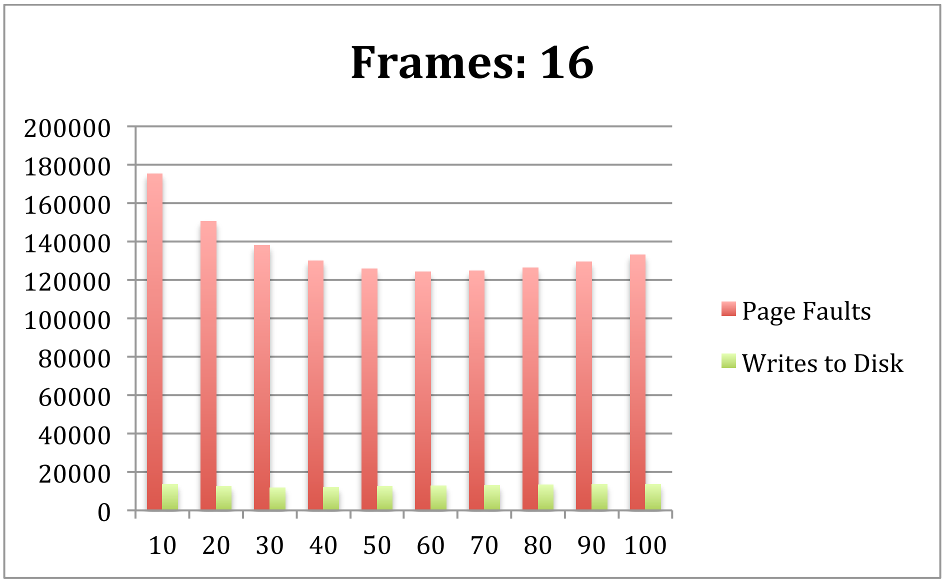
John McQuown

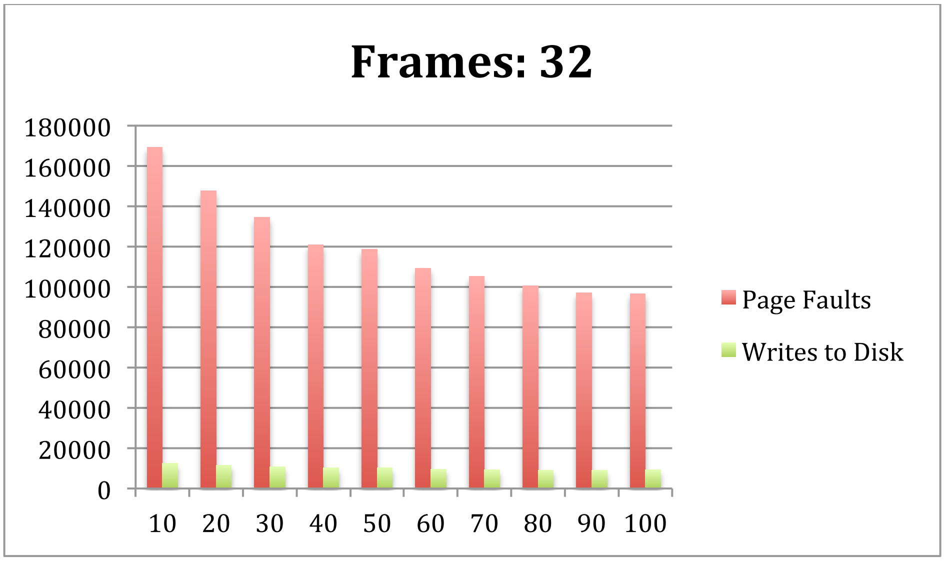
CS 1550 Recitation: Thurs 3:00-3:50 PM

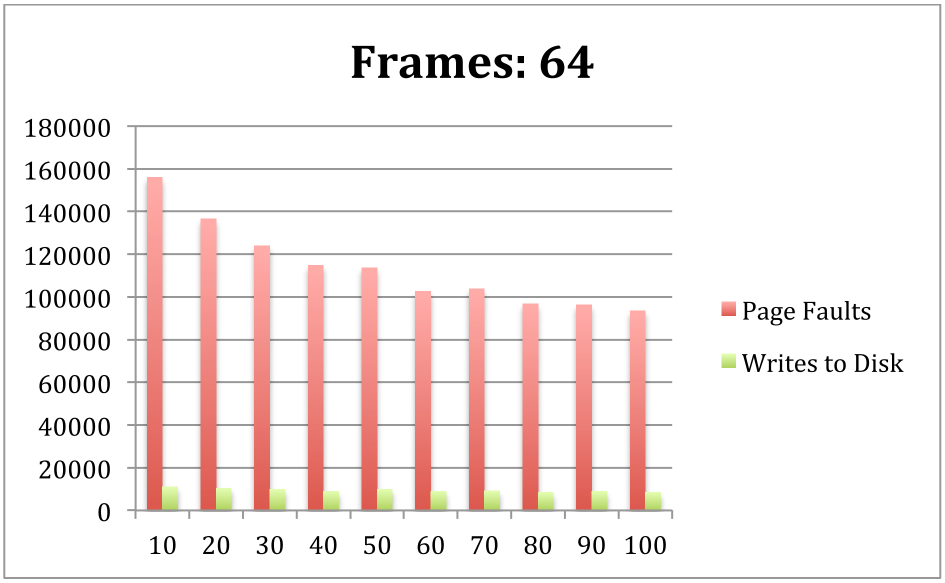
Project 3 Write Up

NRU









Based on these graphs and the data that I have gathered for NRU, the best refresh rate will depend on the amount of frames. In the case of 8 frames, 20 is clearly the best choice as the number of page faults steadily increases as the refresh rate increases. However, in the case of 16, 32, and 64 frames, 60 as a refresh seems to be the best in those three situations. Even though 60 is not the lowest in all cases, it still outputs a low number of pages faults relative to the other refresh rates. For 16, 32, and 64 frames, 60 is below the average number of page faults, which in my opinion makes it a good choice for the best refresh rate.

Other Algorithms

Frames: 8

Algorithm	Page Faults	Writes to Disk
Optimal	118480	15031
Clock	181856	29401
Aging	217376	33221

Frames: 16

Algorithm	Page Faults	Writes to Disk
Optimal	80307	11316
Clock	121682	16376
Aging	209978	30454

Frames: 32

Algorithm	Page Faults	Writes to Disk
Optimal	55802	8274
Clock	87686	12283
Aging	196531	26251

Frames: 64

Algorithm	Page Faults	Writes to Disk
Optimal	38050	5730
Clock	61640	9346
Aging	181258	19133

Based on this data it is clear that Optimal is the best algorithm, but it is of course impossible to actually implement. Clock is consistently the algorithm with the lowest number of page faults and writes to disk, NRU included. Aging however performed very poorly in comparison to the other algorithms. This could have been because of my implementation, as I think that conceptually Aging is a very good algorithm for page replacement. Because of the data, if I were to choose an algorithm for an operating system to use, I would choose clock. I would choose Clock mostly because of its simplicity in terms of implementation and that it is not a very expensive algorithm to run. It is the algorithm with the lowest number of page faults and is the only one with numbers comparable to the Optimal algorithm.