Matthew Viego

Page Replacement Algorithms Report

Four different page replacement algorithms were used in this project. The results of the optimal, clock, aging and working set clock algorithms are displayed by graphs and can be viewed at the bottom so this report. This project required the use of two different files for simulation, so performance will vary depending on the file given for simulation.

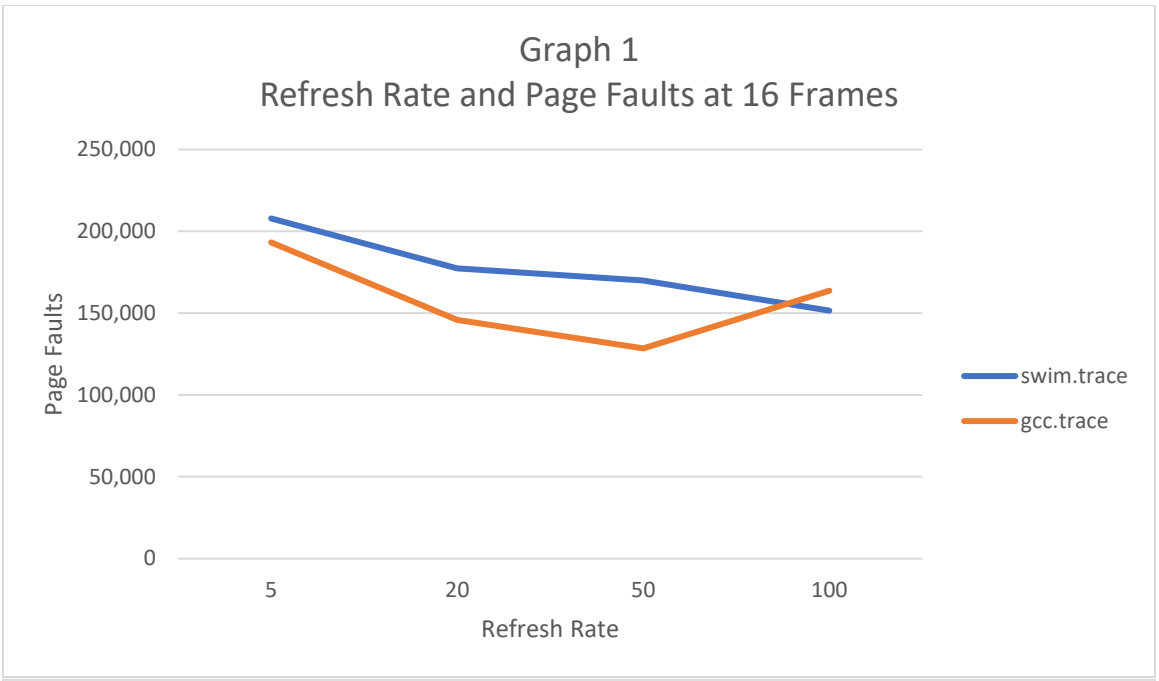
The aging algorithm required the use of a refresh rate. In this algorithm, a refresh rate is chosen which will reset the referenced bit of all frames currently within memory after a certain amount of instructions of occurred. A refresh rate that determines the absolute best results must be chosen. Graph 1 and Graph 2 were created to find this best refresh rate. By looking at the graphs, it is determined that the best and most fair refresh rate to choose is going to be 50. This refresh rate will not provide the aging algorithm with the least amount of page faults, but it will provide a low number of writes before that number shoots up upon raising the refresh rate much higher.

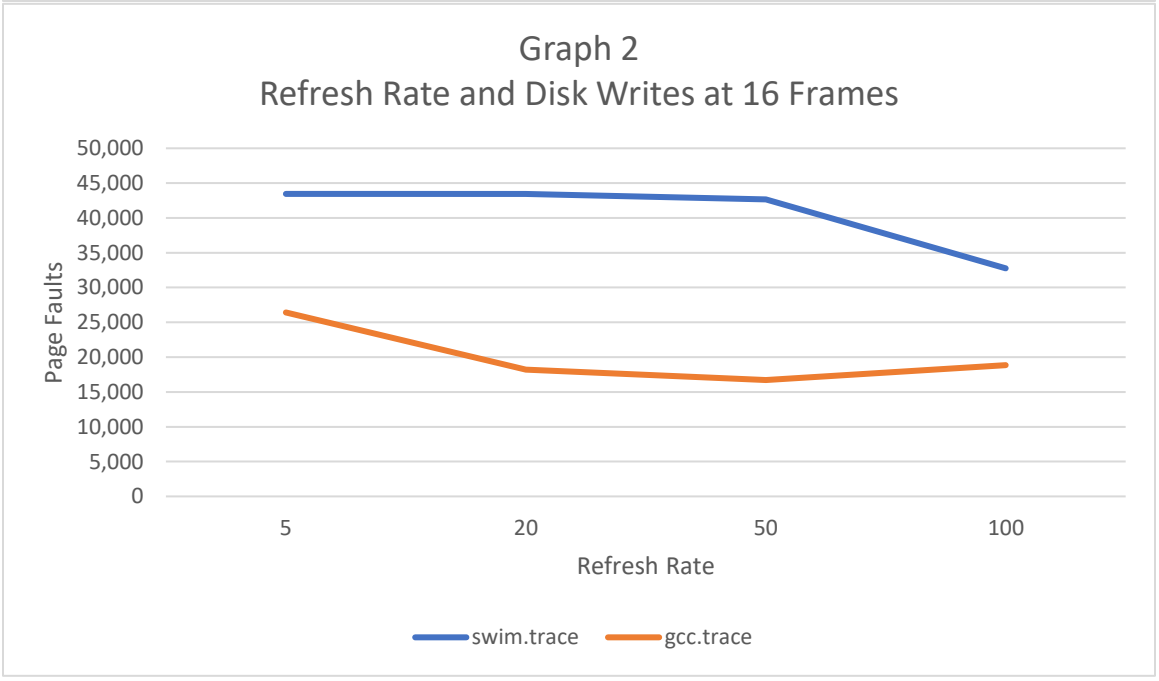
The working set clock algorithm needed to use a tau value. The tau value represented the maximum distance that a page's previous time of use was from the current time. If the page had not been used in a period that was longer than the tau, it was eligible to be evicted. Graphs 3 and 4 below show a test for the best possible tau values. As the tau value went up, page faults went up, but disk writes went down. Just like refresh rate, a happy medium will be the best judgement here. A tau value of 75 will provide an adequate number of page faults with an equivalent number of disk writes.

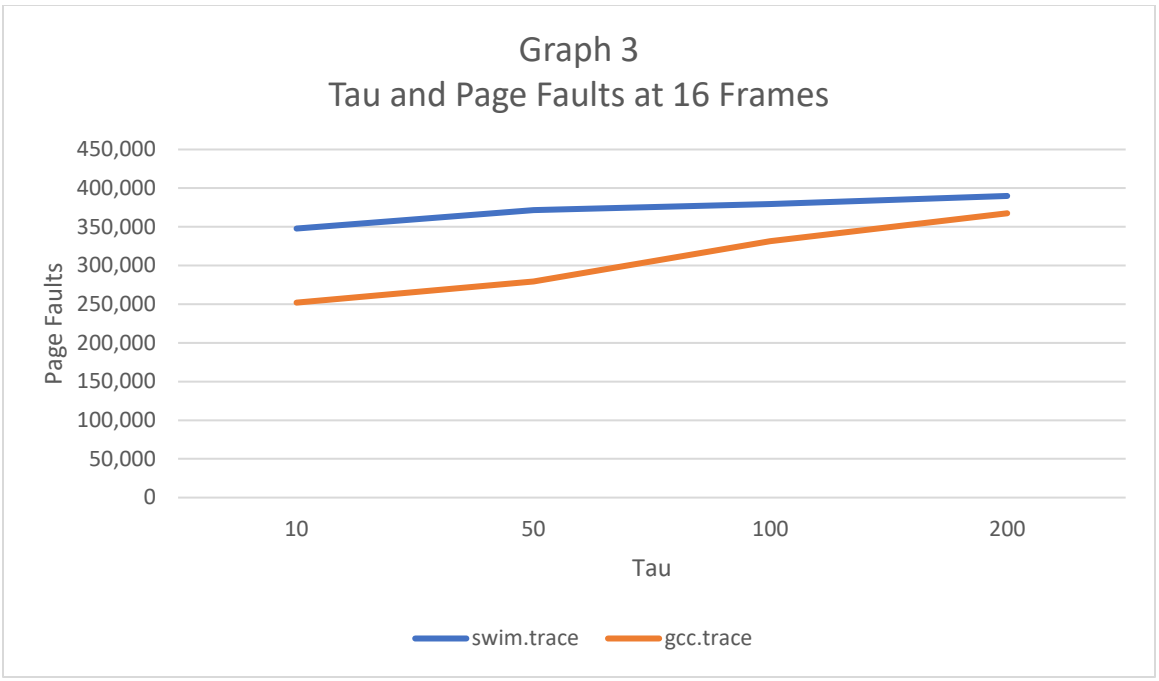
Optimal would clearly be the best algorithm to use. It crushes its competition in every regard. Unfortunately, this algorithm is cannot be implemented because it requires information

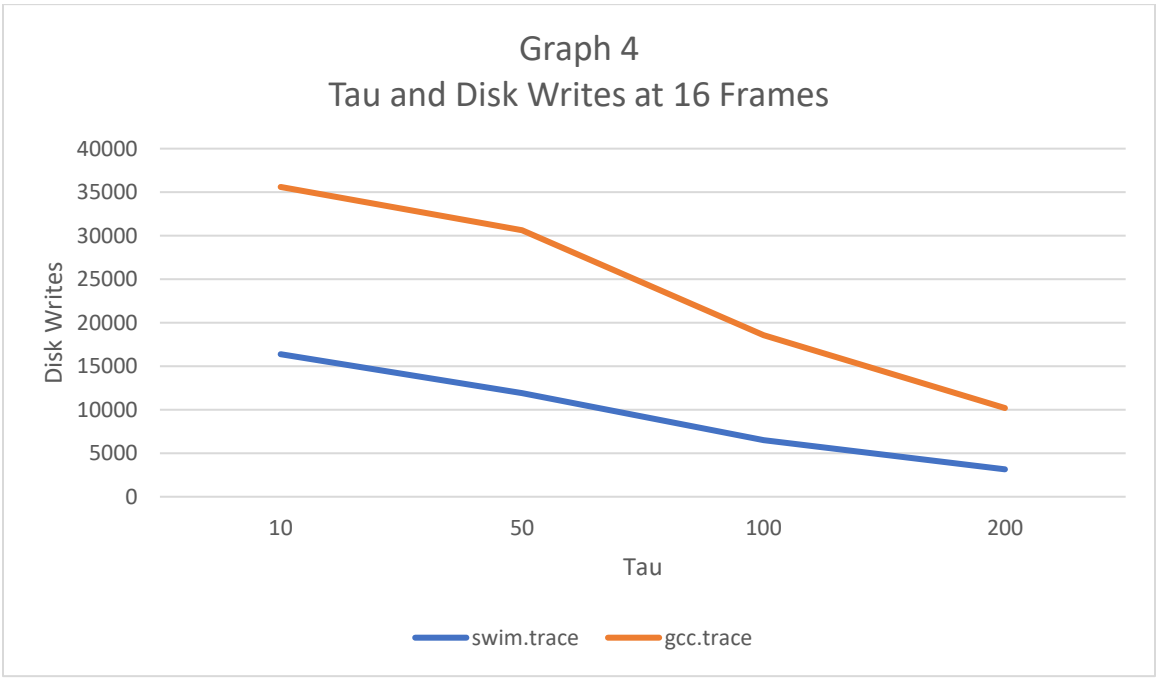
on the future actions of each page. Clock is going to be the best algorithm for an operating system to use. It beats out aging and work in almost every single regard. There are moments when aging pulls a better result than clock such as page faults in swim.trace at the 16 frame mark (Graph 5). The majority of the time, clock beats aging in page faults and disk writes. Working set clock is an interesting algorithm as there are many moments, such as disk writes for swim.trace at the 8 frames mark (Graph 7), where working set clock absolutely annihilates everything else. There is just way too much of a trade off in page faults for this low disk writes. Looking at working set clock for page faults in swim.trace at the 8 frames mark(Graph 5), it is in dead last by a wide margin, especially compared to clock. Working set clock can be fantastic, but only in one aspect at a time. Clock is a better algorithm because there is great parity between page faults and disk writes.

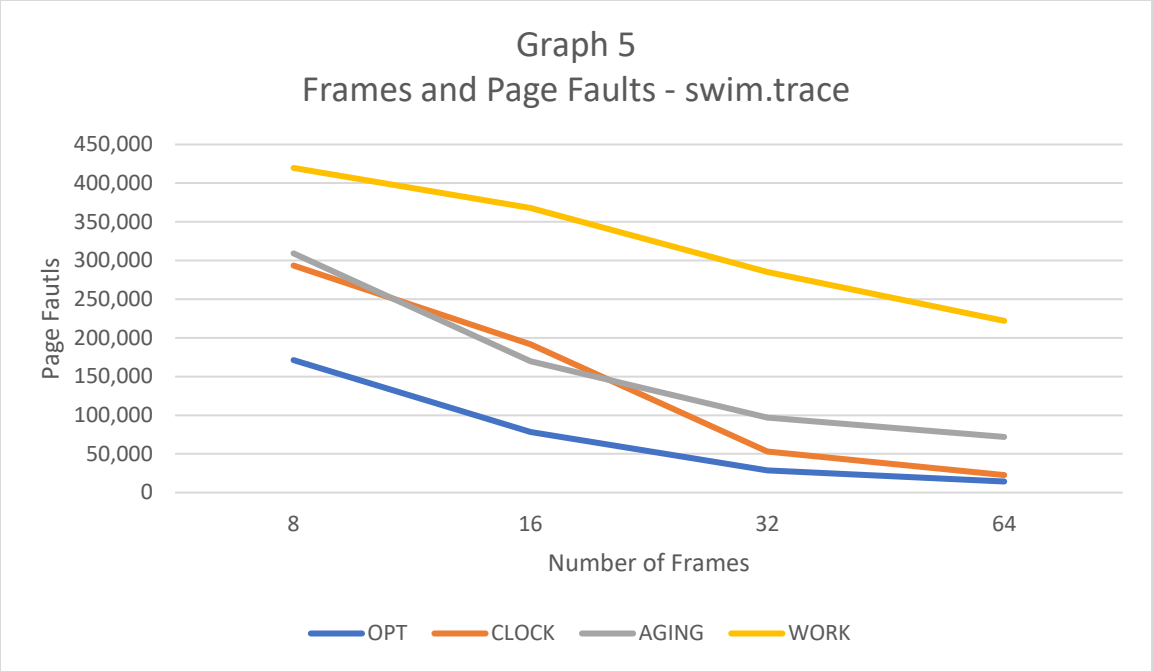
After testing out the data and displaying the information of graphs, it is clear the clock is the best suited algorithm to be implemented into an operating system. Unlike the optimal algorithm, it can actually be implemented. It beats out aging in way more categories than it loses. Clock also does not require intense tradeoffs like working set clock does. Clock is the best algorithm

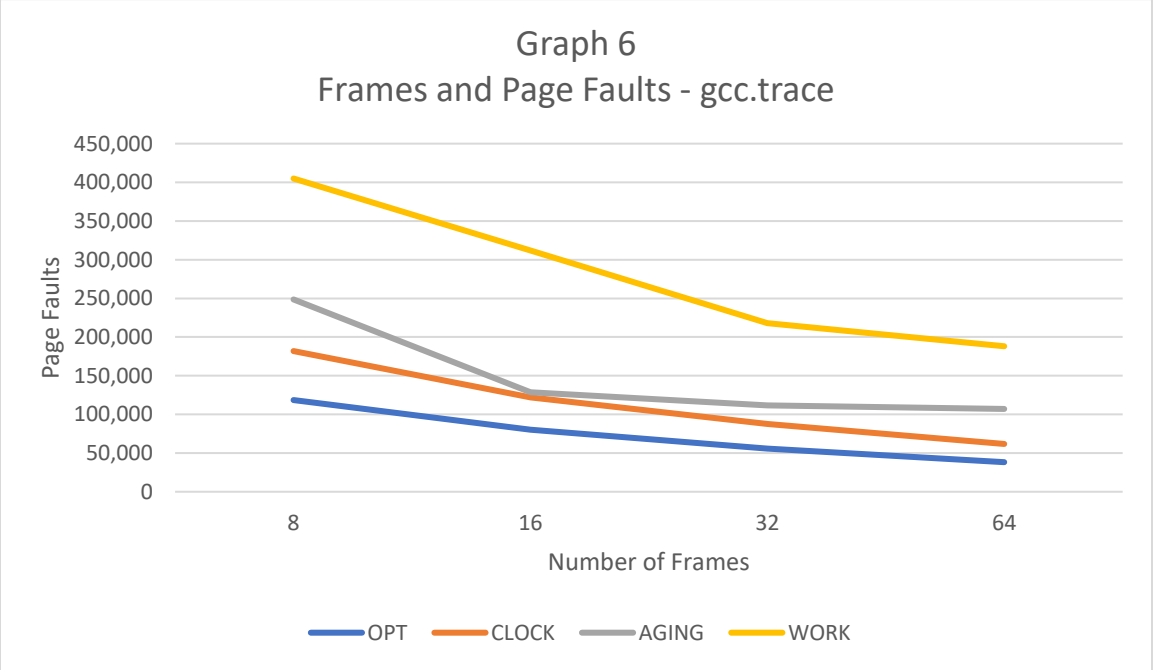

Graph 1
Refresh Rate and Page Faults at 16 Frames
250,000
200,000
150,000
100,000
50,000
0
Page Faults
5
20
50
100
Refresh Rate
swim.trace
gcc.trace

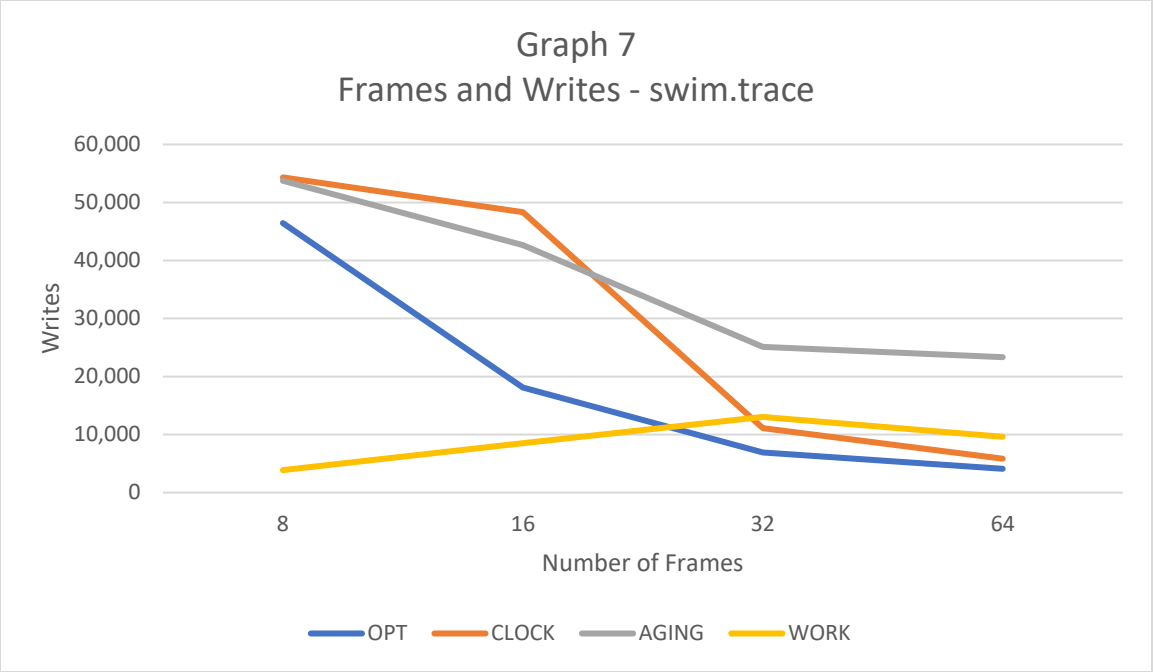

Graph 2
Refresh Rate and Disk Writes at 16 Frames
50,000
45,000
40,000
35,000
30,000
25,000
20,000
15,000
10,000
5,000
0
Page Faults
5
20
50
100
Refresh Rate
swim.trace
gcc.trace

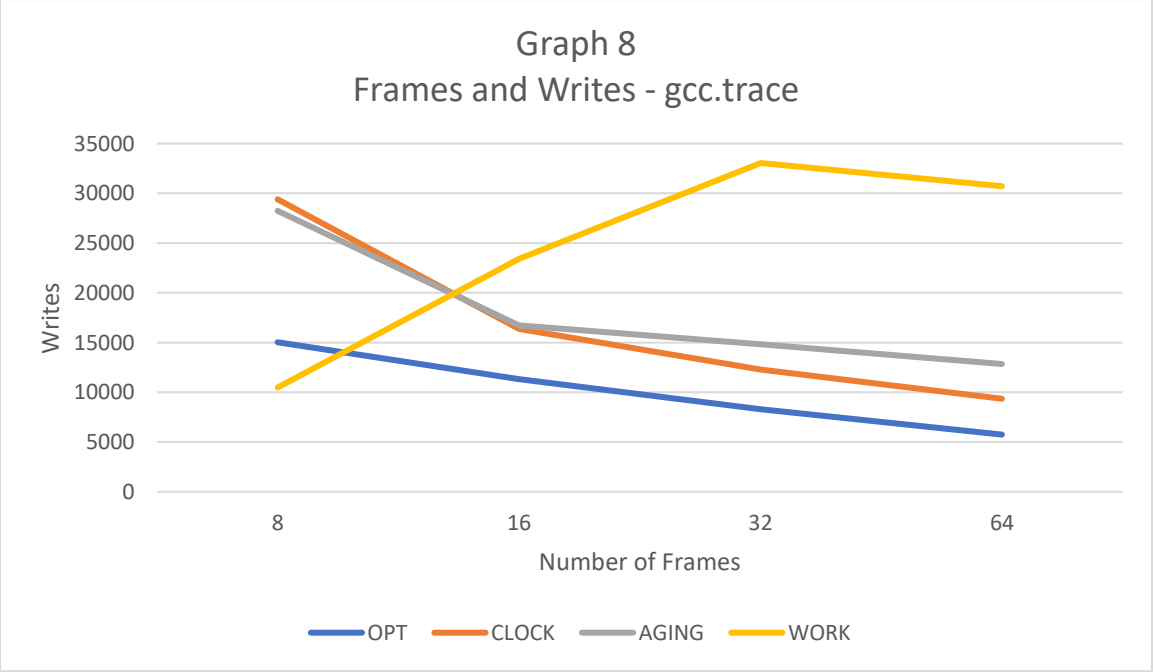

Graph 3
Tau and Page Faults at 16 Frames
450,000
400,000
350,000
300,000
250,000
200,000
150,000
100,000
50,000
0
Page Faults
10
50
100
200
Tau
swim.trace
gcc.trace


Graph 4
Tau and Disk Writes at 16 Frames
40000
35000
30000
25000
20000
15000
10000
5000
0
Disk Writes
10
50
100
200
Tau
swim.trace
gcc.trace


Graph 5
Frames and Page Faults - swim.trace
450,000
400,000
350,000
300,000
250,000
200,000
150,000
100,000
50,000
0
Page Faults
8
16
32
64
Number of Frames
OPT
CLOCK
AGING
WORK


Graph 6
Frames and Page Faults - gcc.trace
450,000
400,000
350,000
300,000
250,000
200,000
150,000
100,000
50,000
0
Page Faults
8
16
32
64
Number of Frames
OPT
CLOCK
AGING
WORK


Graph 7
Frames and Writes - swim.trace
60,000
50,000
40,000
30,000
20,000
10,000
0
Writes
8
16
32
64
Number of Frames
OPT
CLOCK
AGING
WORK


Graph 8
Frames and Writes - gcc.trace
35000
30000
25000
20000
15000
10000
5000
0
Writes
8
16
32
64
Number of Frames
OPT
CLOCK
AGING
WORK