# Project 3 Results

Christian Boni

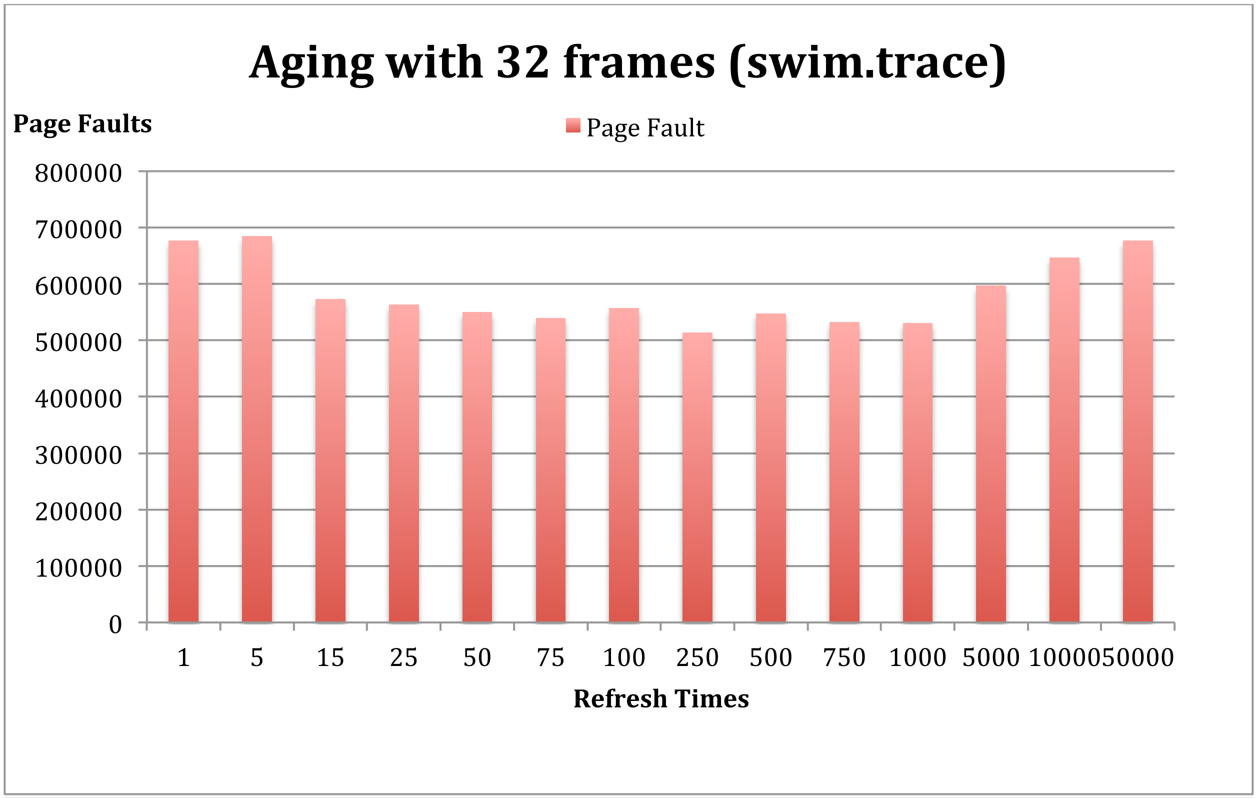
CS 1550

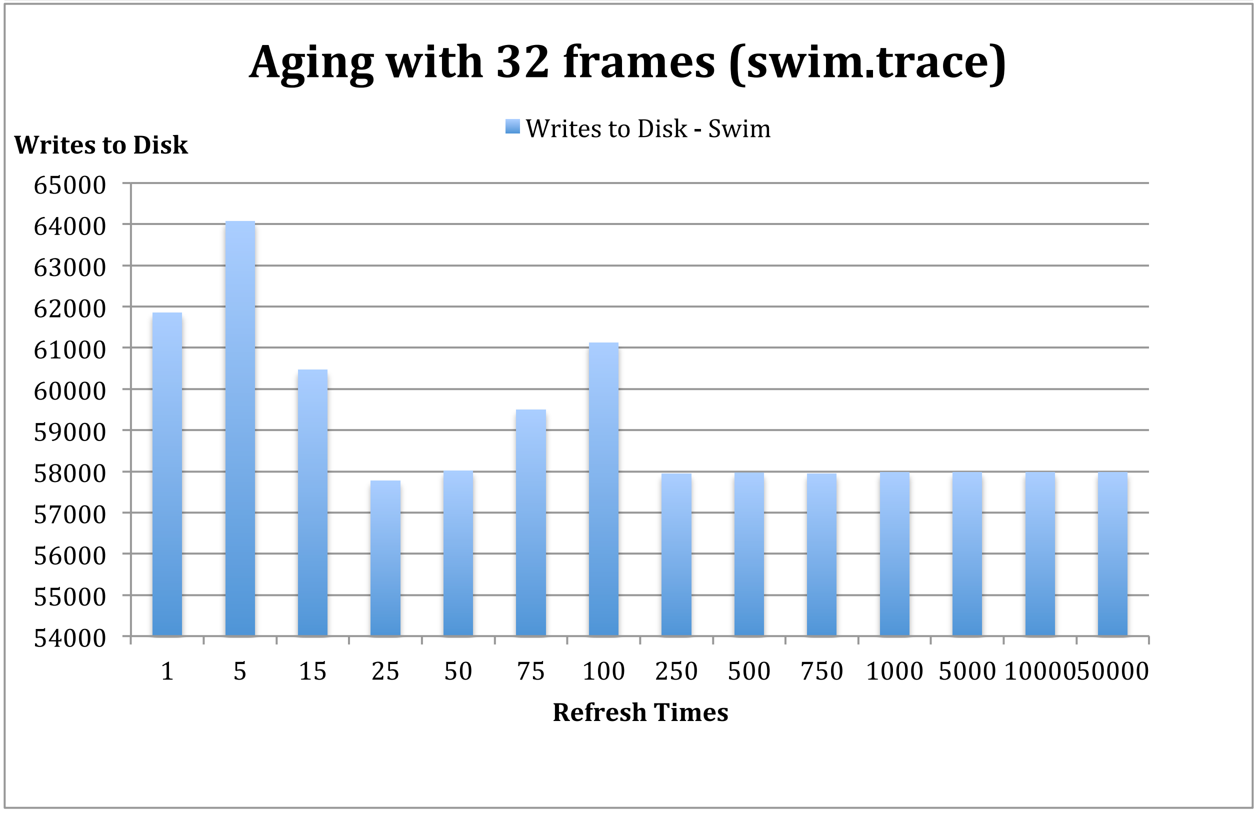
# Aging

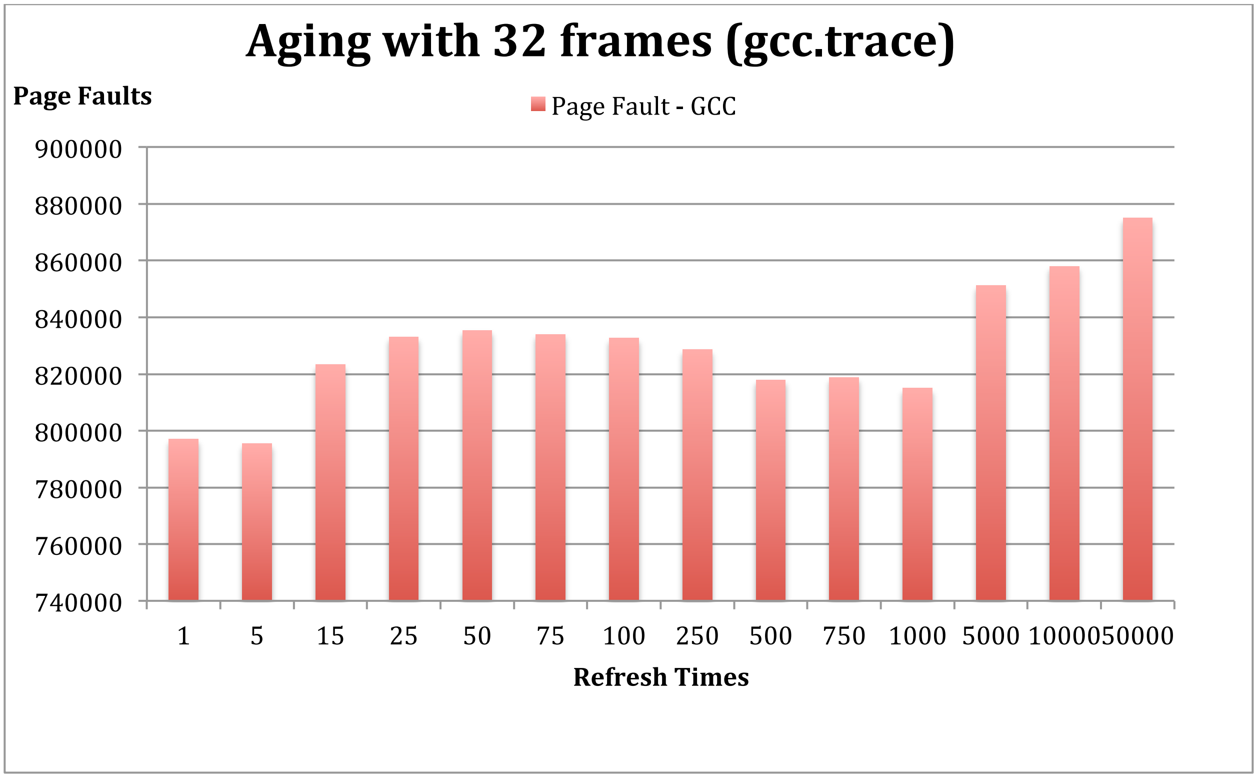
First we will observe the results from varying refresh times on the aging algorithm with 32 frames and the swim trace file. Looking at the first graph of refresh time versus the number of page faults it is clearly shown that the graph forms a parabolic curve.  Both smaller values and larger values for the refresh time cause more page faults. However, this is not the case for the graph with refresh time versus the number of writes to disk. It seems to be that as the refresh time increases the number of writes to disk tends to decrease until it reaches a minimum value around the refresh time of 250 where it stops decreasing altogether.

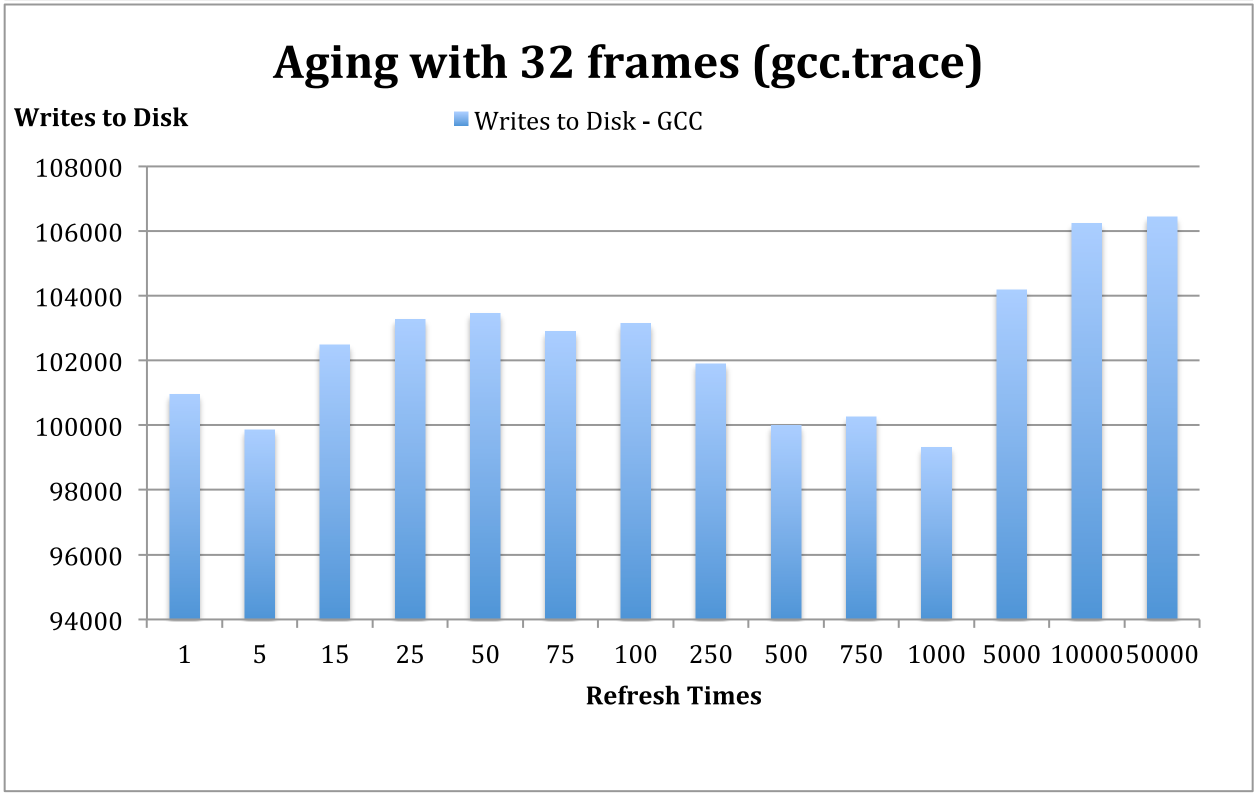
Now, looking at the graphs varying refresh times on the aging algorithm with 32 frames and the gcc trace file things change slightly. Very small refresh times cause both the number of page faults and number of writes to disk show to be very low. This behavior could possibly be considered as outliers. As the refresh time increases it begins to show similar behavior to the swim trace's graph of refresh time versus the number of page faults. While the refresh time increases the number of page faults decreases up to a certain point when it starts increasing again, causing a slightly parabolic curve as the other graph portrayed. Now the next graph of refresh time versus the number of writes to disk also follows this trend, which is different from the swim trace's graph.

While observing all of the data given from this algorithm it seems to be that maybe some of the Aging algorithm's downfalls may be causing this algorithm to perform strangely. A couple of these downfalls could be the 8 bit counter not being able to hold as many ticks/past references as a 16 or 32 bit counter would, or that a newly added page's counter of all 0's cannot be distinguished from a page that has been in memory for a very long time. In conclusion, after observing all of the data I found that a good refresh time was around the 250-500 area. In my case I went with the refresh time of 250 to use while testing the WSClock algorithm.


Aging with 32 frames (swim.trace)
Page Faults
🟥 Page Fault
800000
700000
600000
500000
400000
300000
200000
100000
0
1
5
15
25
50
75
100
250
500
750
1000
5000
10000
50000
Refresh Times


Aging with 32 frames (swim.trace)
Writes to Disk
🟦 Writes to Disk - Swim
65000
64000
63000
62000
61000
60000
59000
58000
57000
56000
55000
54000
1
5
15
25
50
75
100
250
500
750
1000
5000
10000
50000
Refresh Times

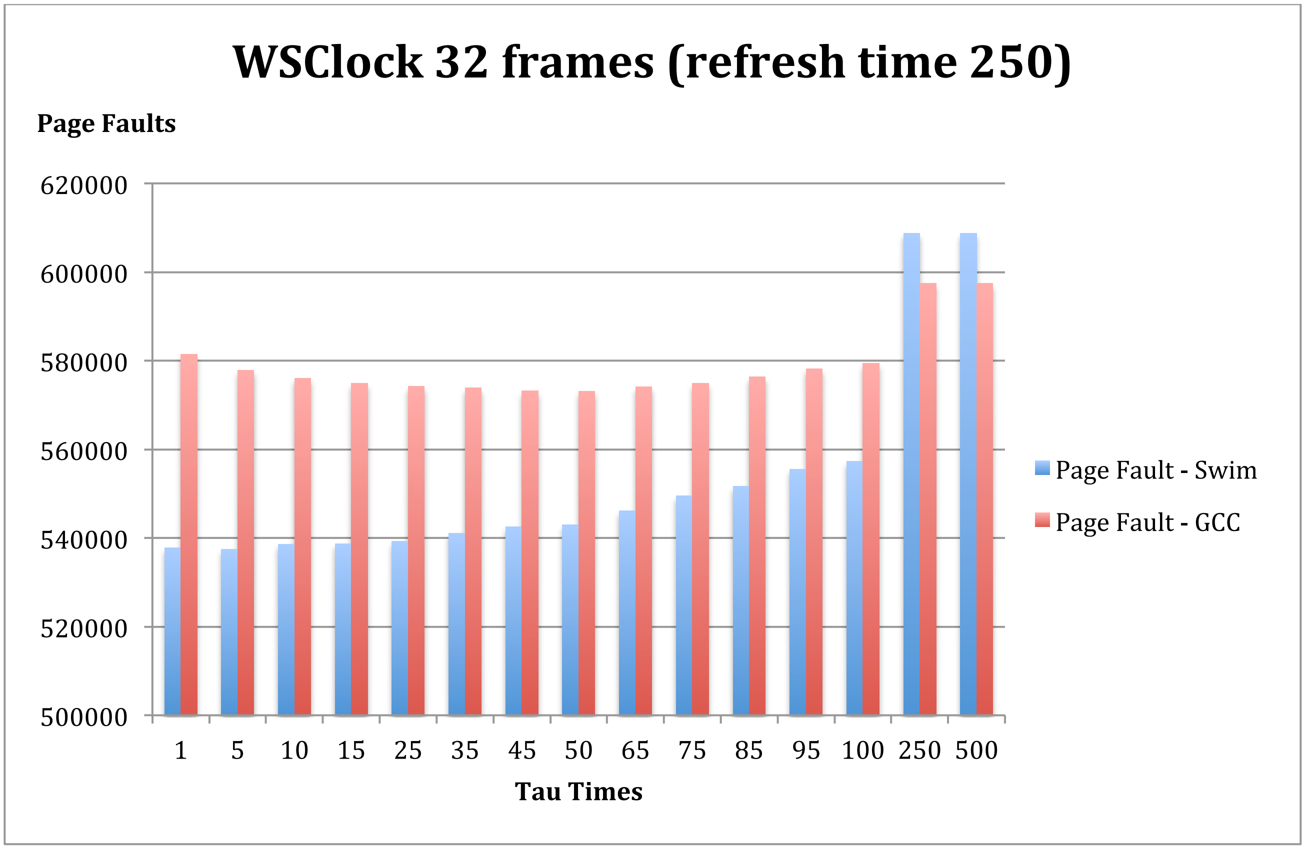

Aging with 32 frames (gcc.trace)
Page Faults
Page Fault - GCC
900000
880000
860000
840000
820000
800000
780000
760000
740000
1
5
15
25
50
75
100
250
500
750
1000
5000
10000
50000
Refresh Times

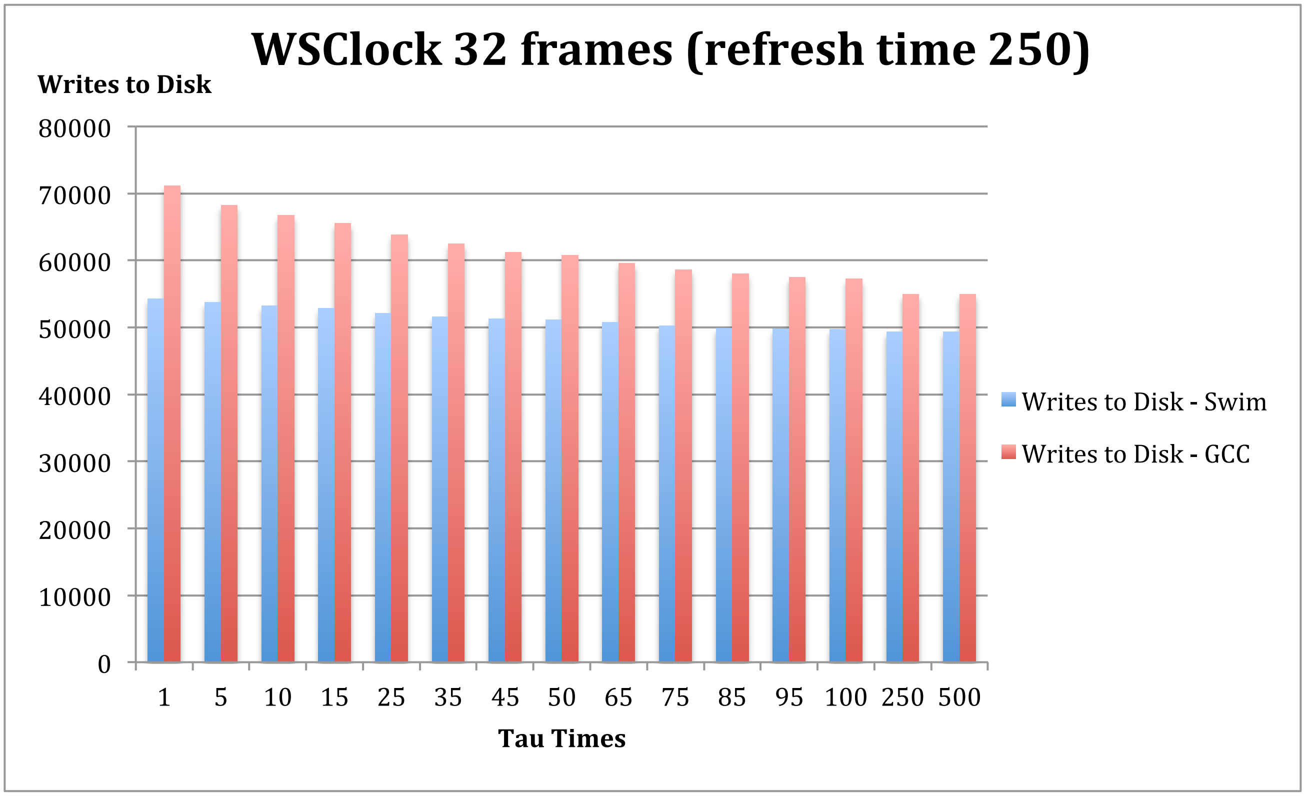

Aging with 32 frames (gcc.trace)
Writes to Disk
Writes to Disk - GCC
108000
106000
104000
102000
100000
98000
96000
94000
1
5
15
25
50
75
100
250
500
750
1000
5000
10000
50000
Refresh Times

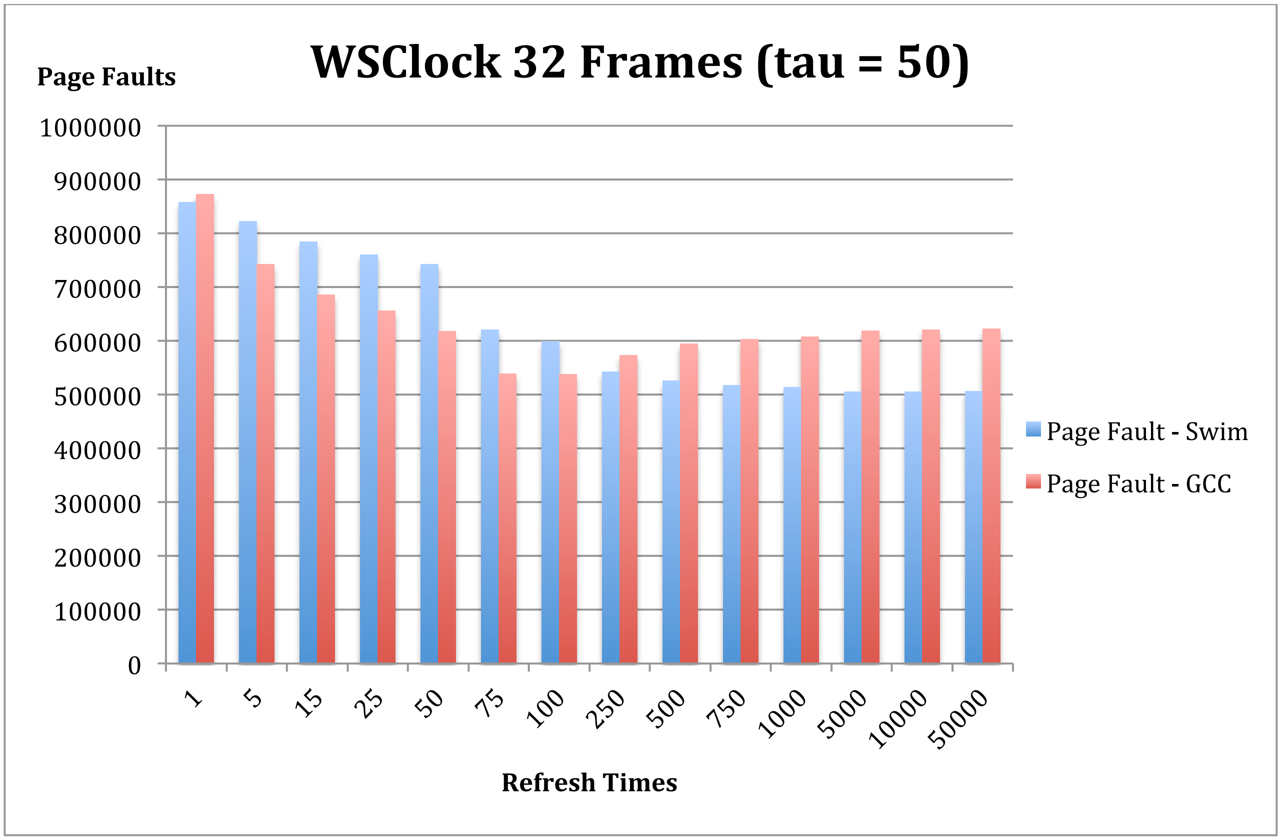
# WSClock

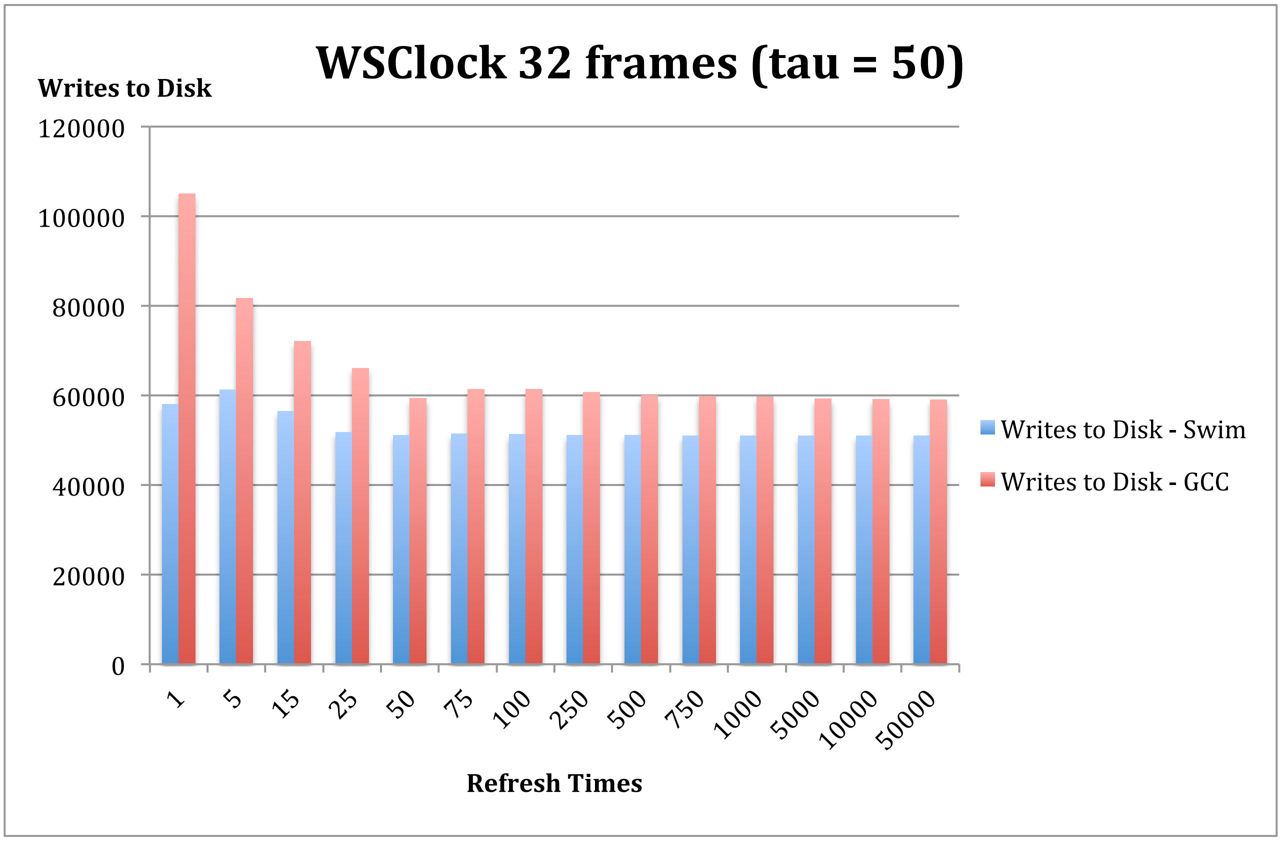
We now can observe the results from the WSClock algorithm with 32 frames and a refresh time of 250 with both the swim and gcc trace files while varying tau.
We will look at the first graph of tau versus the number of page faults first for both the swim and gcc trace files. In the case of the gcc trace file, varying tau creates a sort of parabolic curve where both low values and high for tau cause more page faults. However, when tau reaches a value of 250 the page faults reach a maximum and stop increasing altogether. On the other hand, the swim trace file shows that as tau increases so does the number of page faults, up until the 250-point, where again the number of page faults reaches a maximum. Now we will look at the second graph of tau versus the number of writes to disk for both the swim and gcc trace files. This graph is clear as it shows for both trace files that as tau increases the number of writes to disk decreases. This trend is remains true until a minimum of writes to disk is reached when tau is in the 250 to 500 ranges. As a result, higher values for tau causes less writes to disk and in general lower values for tau causes less page faults. That being known the value for tau that I chose was 50 since it was in the middle of the graph which minimized both the number of page faults and writes to disk.

Shifting our attention to the results from the WSClock algorithm with 32 frames and a tau of 50 with both the swim and gcc trace files while varying refresh time. We will look at the first graph of the refresh time versus the number of page faults first for both the swim and gcc trace files. Again in the case of the gcc trace file, varying refresh time creates a sort of parabolic curve where both low values and high for refresh time cause more page faults. Although, when refresh time reaches a value of 5000 the page faults reach a maximum and stop increasing altogether. On the other hand, the swim trace file shows that as refresh time increases the number of page faults decreases, up until the 250-point, where the number of page faults reaches a minimum. Now we will look at the second graph of refresh time versus the number of writes to disk for both the swim and gcc trace files. Again this graph is clear as it shows for both trace files that as refresh time increases the number of writes to disk decreases. However, this trend doesn't hold for long. The minimum number of writes to disk is reached when refresh time is in the 25 to 50 ranges. As a result, higher values for refresh time causes less writes to disk and in general values for refresh time in the middle of the graph causes less page faults. That being known I chose to keep my original refresh time of 250 since it was in the middle of the graph which minimized both the number of page faults and writes to disk.


WSClock 32 frames (refresh time 250)
Page Faults
620000
600000
580000
560000
540000
520000
500000
1
5
10
15
25
35
45
50
65
75
85
95
100
250
500
Tau Times
Page Fault - Swim
Page Fault - GCC

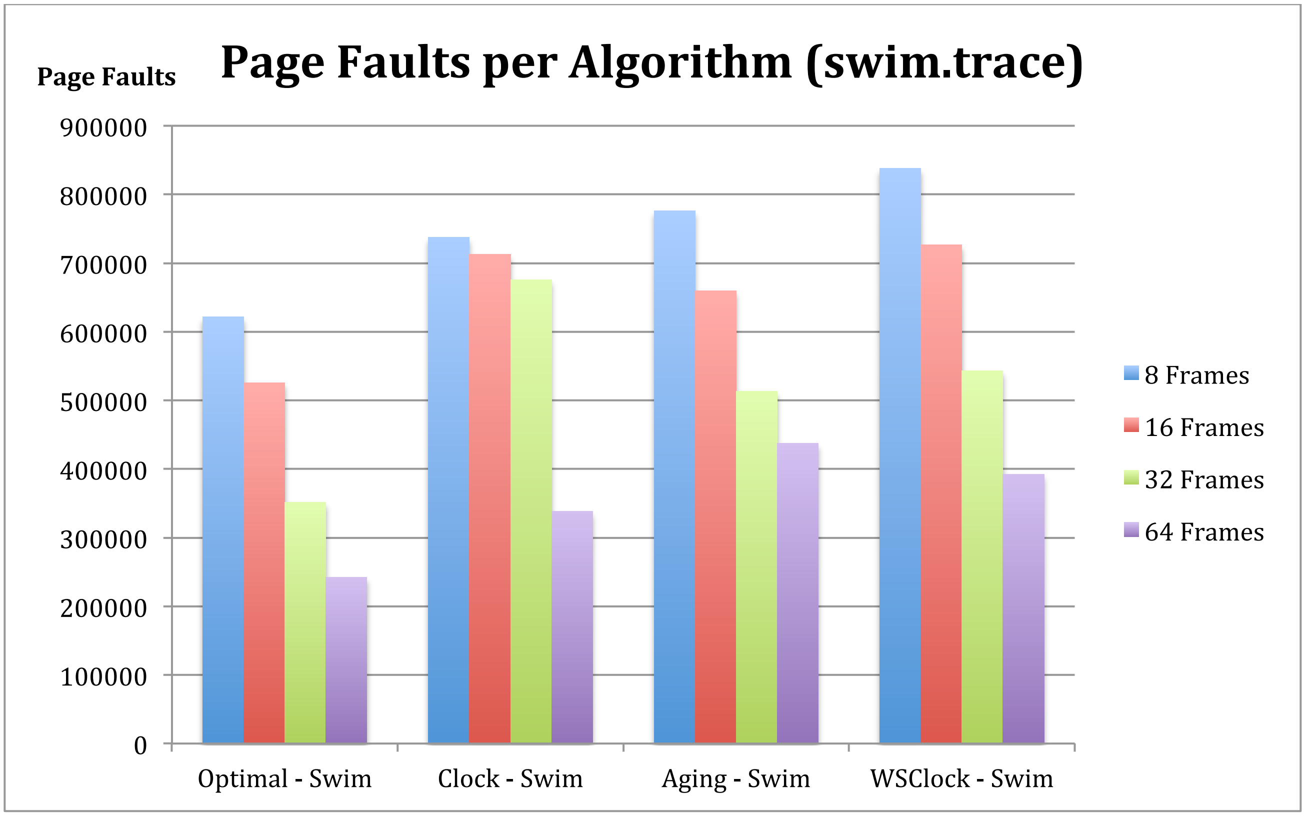

WSClock 32 frames (refresh time 250)
Writes to Disk
80000
70000
60000
50000
40000
30000
20000
10000
0
1
5
10
15
25
35
45
50
65
75
85
95
100
250
500
Tau Times
Writes to Disk - Swim
Writes to Disk - GCC

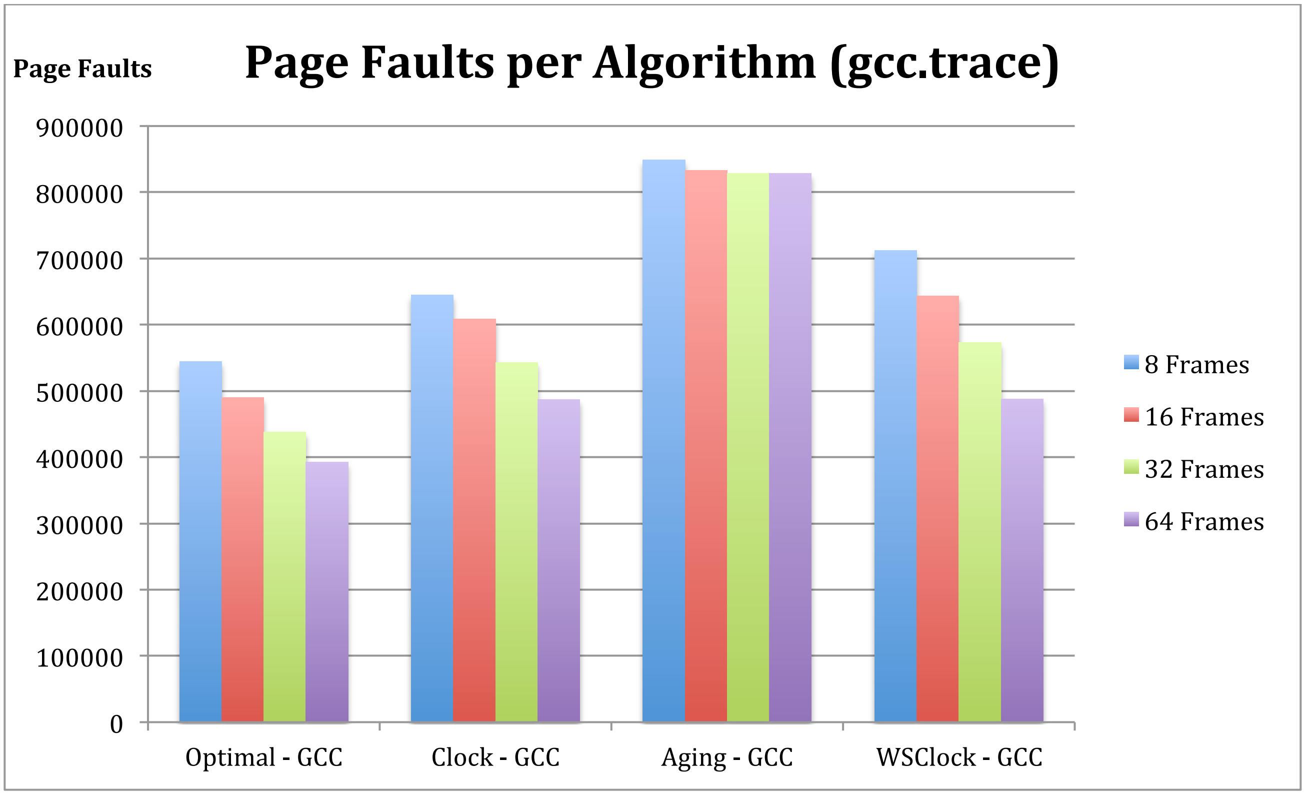

WSClock 32 Frames (tau = 50)
Page Faults
1000000
900000
800000
700000
600000
500000
400000
300000
200000
100000
0
1
5
15
25
50
75
100
250
500
750
1000
5000
10000
50000
Refresh Times
Page Fault - Swim
Page Fault - GCC

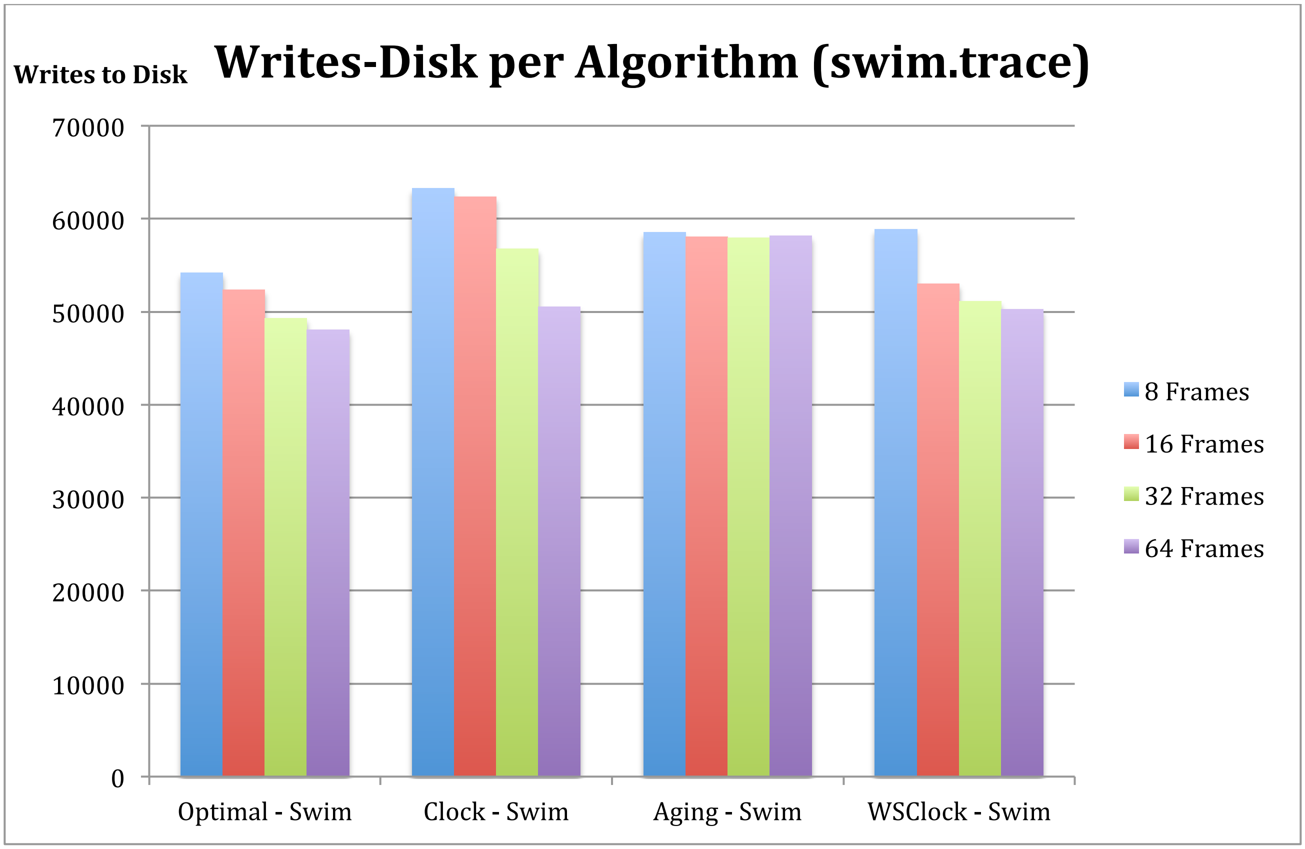

WSClock 32 frames (tau = 50)
Writes to Disk
120000
100000
80000
60000
40000
20000
0
1
5
15
25
50
75
100
250
500
750
1000
5000
10000
50000
Refresh Times
Writes to Disk - Swim
Writes to Disk - GCC

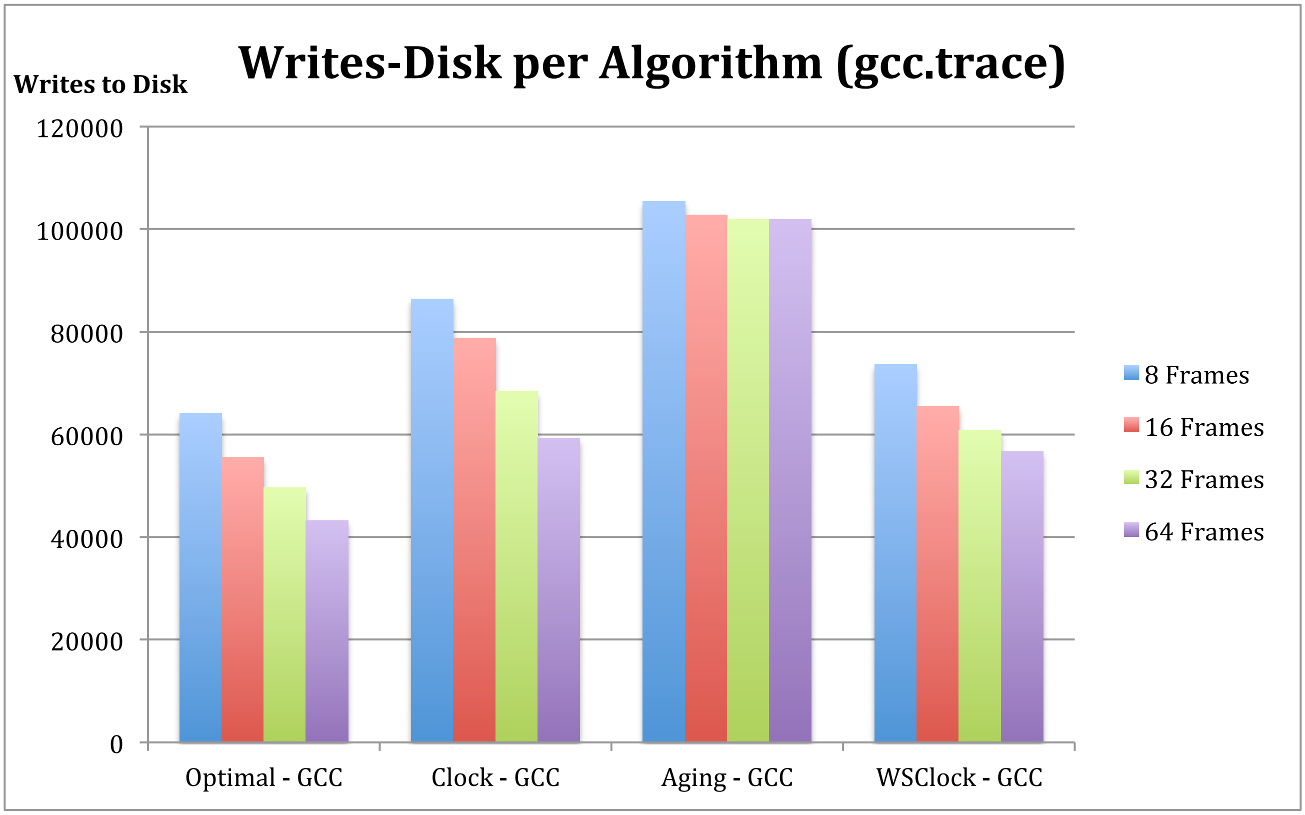
# Conclusion

Finally we can now observe all of the algorithms side by side. First noting that aside from Belady's anomaly we can assure that as the number of frames in memory increases the number of page faults and writes to disk will decrease. Furthermore, we will be comparing our results to the optimal algorithms since it always produces the least amount of page faults and writes to disk. Now when looking at the graph of page faults per algorithm for the swim trace file we can see that both the aging and wsclock implementations perform similarly in that as the number of frames increases the number of page faults drop at about the same rate as the Optimal algorithm. It can be seen that the aging algorithm and wsclock algorithm perform almost identically here. Now looking at the clock algorithm its performance doesn't change much until it has 64 frames in memory. This behavior could potentially be a special case for the clock algorithm in this specific trace file and it may be noted that this algorithm may not always perform like this. Shifting our attention to the graph of page faults per algorithm for the gcc trace file we can see that now both the Clock and wsclock implementations perform similarly in that as the number of frames increases the number of page faults drop at about the same rate as the Optimal algorithm. Now in the case of this trace file the Aging algorithm performs terribly and the number of page faults doesn't decrease much or at all as the number of frames increases. . Also with the clocks behavior in the first graph this could potentially be a special case for the aging algorithm in this specific trace file and it may be noted that this algorithm may not always perform like this. Next observing the final two graphs for the writes to disk per algorithm for the both swim and gcc files we see very similar behavior for both trace files for each algorithm. For both the wsclock and clock algorithm the number of writes to disk steadily decreases as the number of frames in memory increases similar to that of the optimal algorithm. Furthermore, it can be seen that the wsclock algorithm almost reaches the optimal number of page faults for both trace files. Now on the other hand while looking at the aging algorithm for both trace files the number of writes to disk almost remains unchanged and very far away from the optimal amount. In conclusion after observing all of the algorithms side by side and comparing all the data, my choice for the best algorithm to use would be the wsclock algorithm. I chose wsclock since its performance remained stable no matter which trace file was used. Furthermore, the wsclock's data showed that for a majority of the tests its results were not all that far off from the optimal performance. Finally, even though other algorithms may have had less page faults or writes to disk in some cases, the wsclock algorithm showed to be the most consistent and reliable.


Page Faults
Page Faults per Algorithm (swim.trace)
900000
800000
700000
600000
500000
400000
300000
200000
100000
0
Optimal - Swim
Clock - Swim
Aging - Swim
WSClock - Swim
8 Frames
16 Frames
32 Frames
64 Frames


Page Faults
Page Faults per Algorithm (gcc.trace)
900000
800000
700000
600000
500000
400000
300000
200000
100000
0
Optimal - GCC
Clock - GCC
Aging - GCC
WSClock - GCC
8 Frames
16 Frames
32 Frames
64 Frames


Writes to Disk
Writes-Disk per Algorithm (swim.trace)
70000
60000
50000
40000
30000
20000
10000
0
Optimal - Swim
Clock - Swim
Aging - Swim
WSClock - Swim
8 Frames
16 Frames
32 Frames
64 Frames


Writes to Disk
Writes-Disk per Algorithm (gcc.trace)
120000
100000
80000
60000
40000
20000
0
Optimal - GCC
Clock - GCC
Aging - GCC
WSClock - GCC
8 Frames
16 Frames
32 Frames
64 Frames