CS 342
Project 2

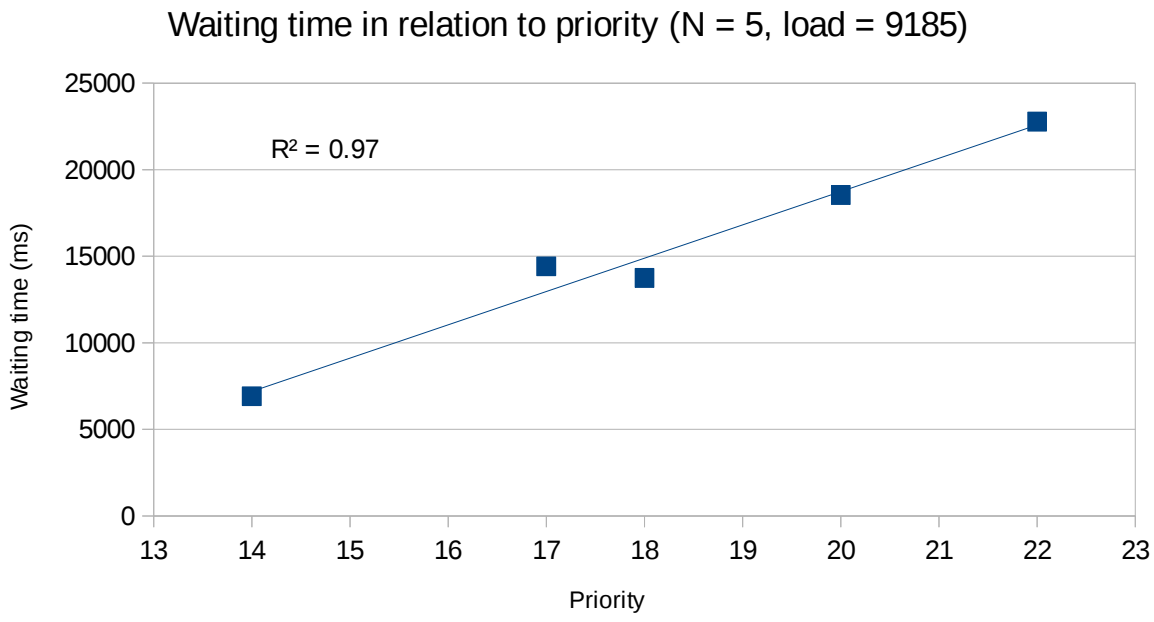
Ata Deniz Aydın
21502637
CS 342-01

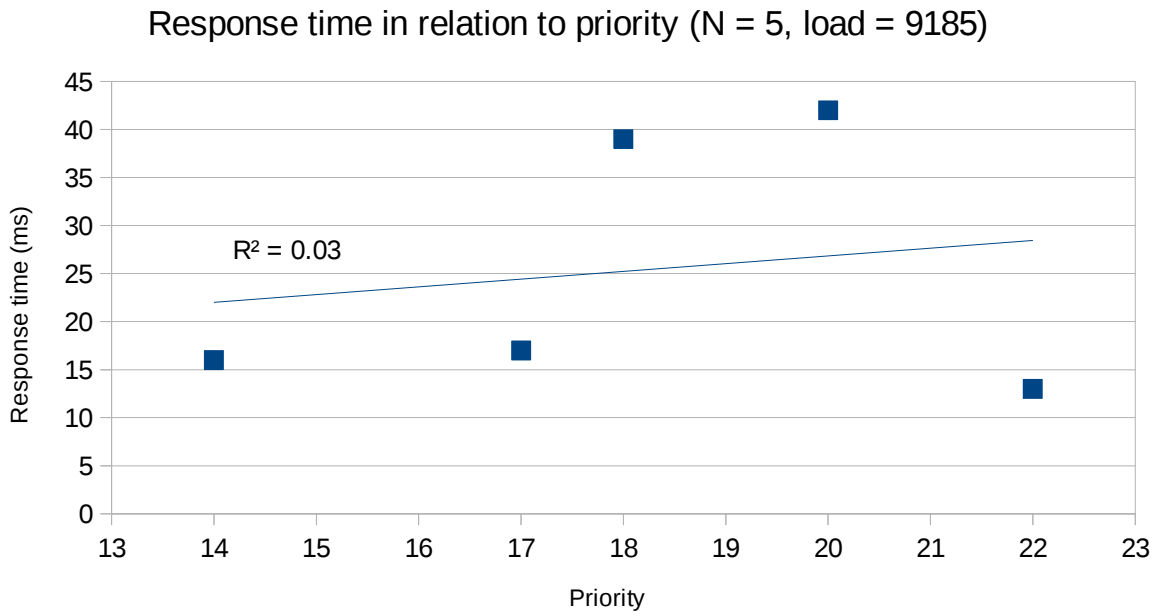
10 workload files each were generated for N (number of processes) = 5, 10 and 20, with the average number of bursts, average starting time, and the average duration of CPU and I/O bursts fixed to 10, 100 ms, 400 ms and 100 ms respectively. The CFS simulator was executed for each workload file, and the priority, turnaround time, total waiting time and average response time of each process were tabulated. For each value of N, the waiting and response times of each process were plotted with respect to the priority of each process. Moreover, the average waiting and response time of processes were plotted with respect to N, as well as to the total load for fixed N.

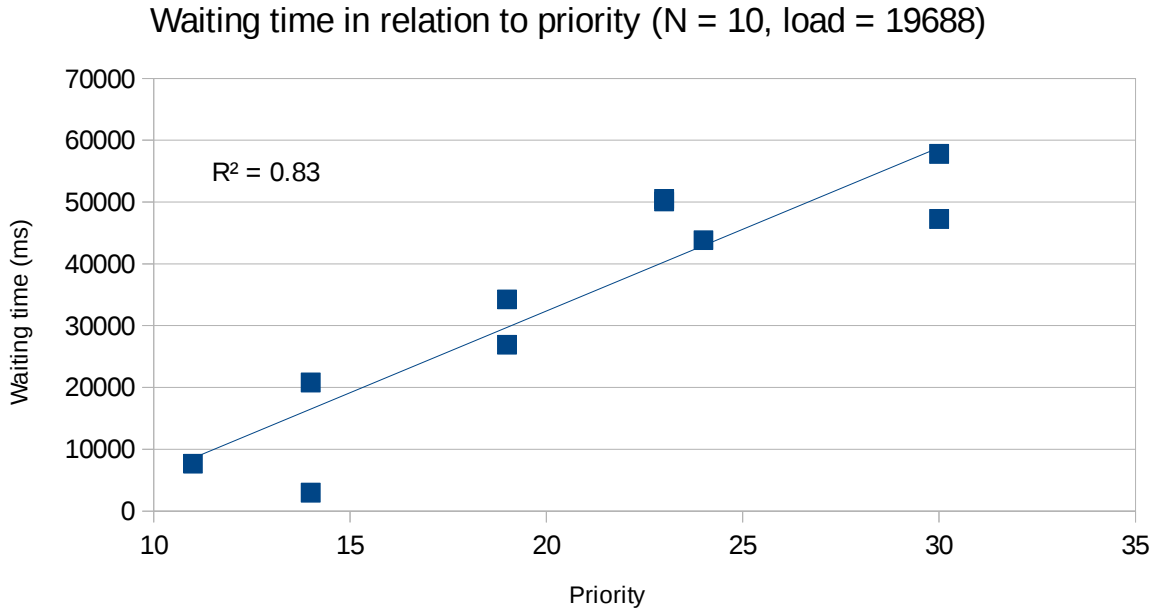
The programs were compiled using GCC 5.4.0 on Ubuntu 16.04.4, running on a VM, with a 64-bit 1.8 GHz Intel Core i5 processor and 4 GB RAM.

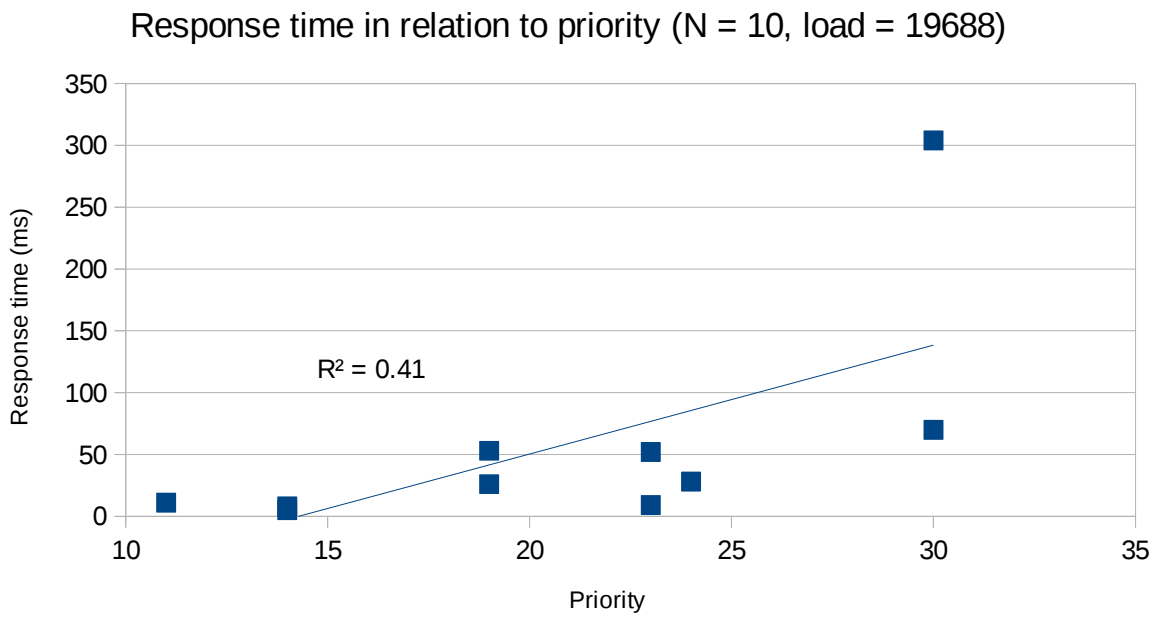
1. Waiting and response times of each process in relation to priority for fixed load

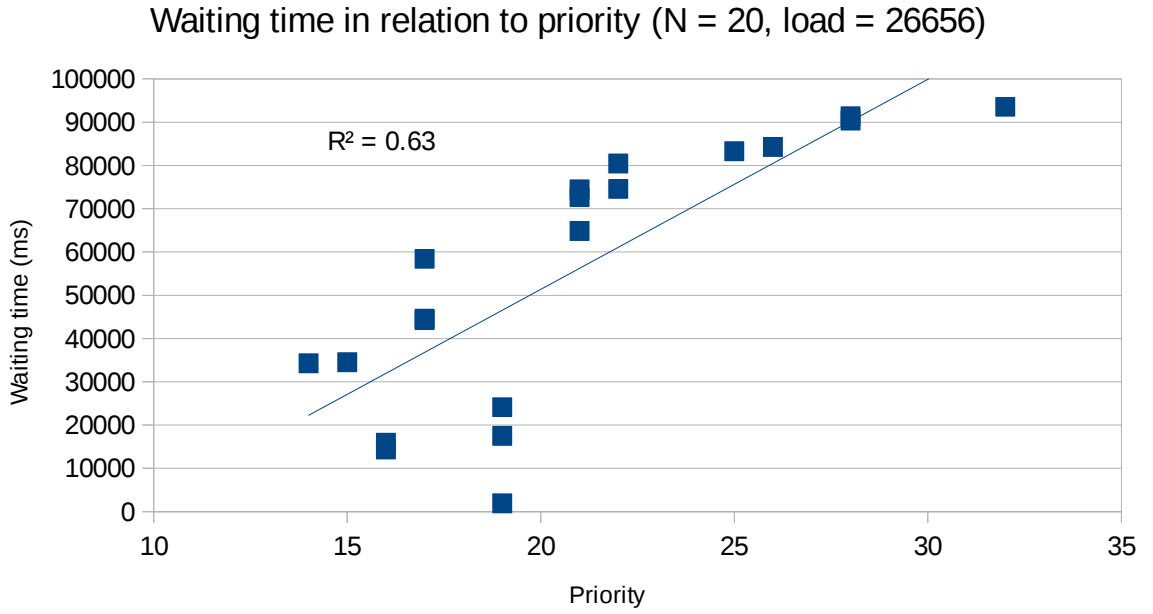
For each value of N, one workload file was selected and simulated. The total waiting times and average response times of each process in each workload file are plotted below.

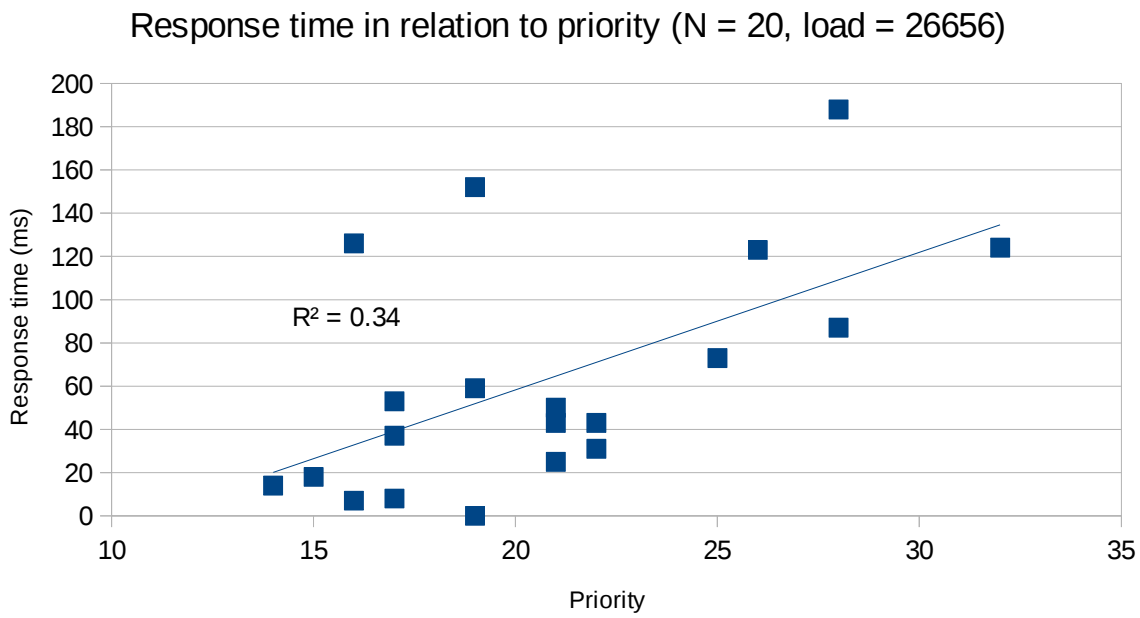



Response time in relation to priority (N = 5, load = 9185)
Response time (ms)
Priority
R² = 0.03
45
40
35
30
25
20
15
10
5
0
13
14
15
16
17
18
19
20
21
22
23


Waiting time in relation to priority (N = 10, load = 19688)
Waiting time (ms)
Priority
R² = 0.83
70000
60000
50000
40000
30000
20000
10000
0
10
15
20
25
30
35


Response time in relation to priority (N = 10, load = 19688)
Response time (ms)
350
300
250
200
150
100
50
0
10
15
20
25
30
35
Priority
R² = 0.41


Waiting time in relation to priority (N = 20, load = 26656)
Waiting time (ms)
100000
90000
80000
70000
60000
50000
40000
30000
20000
10000
0
10
15
20
25
30
35
Priority
R² = 0.63

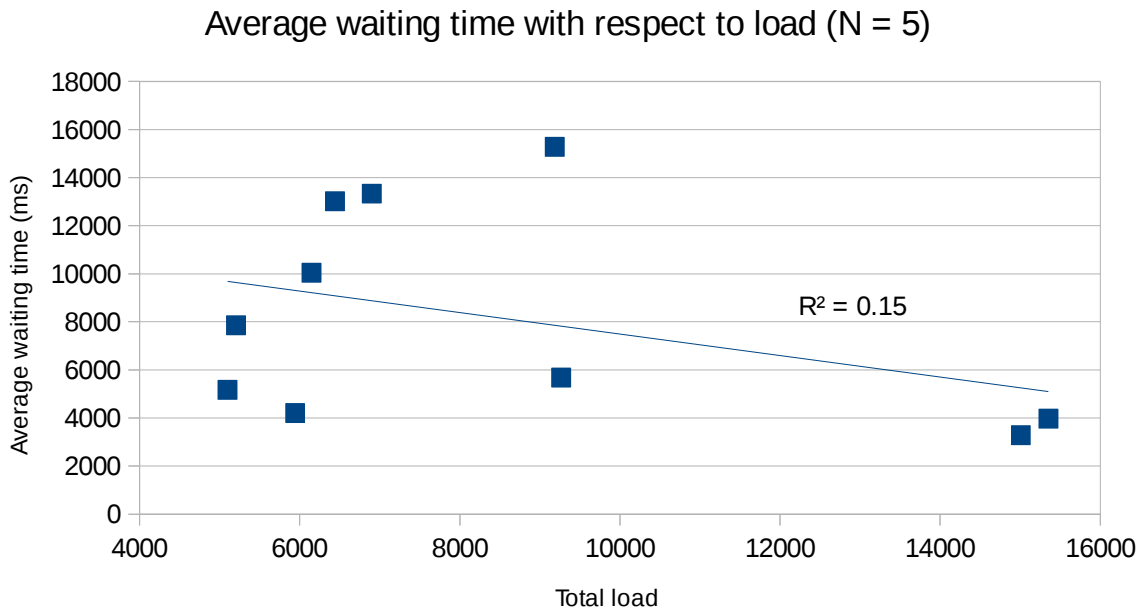


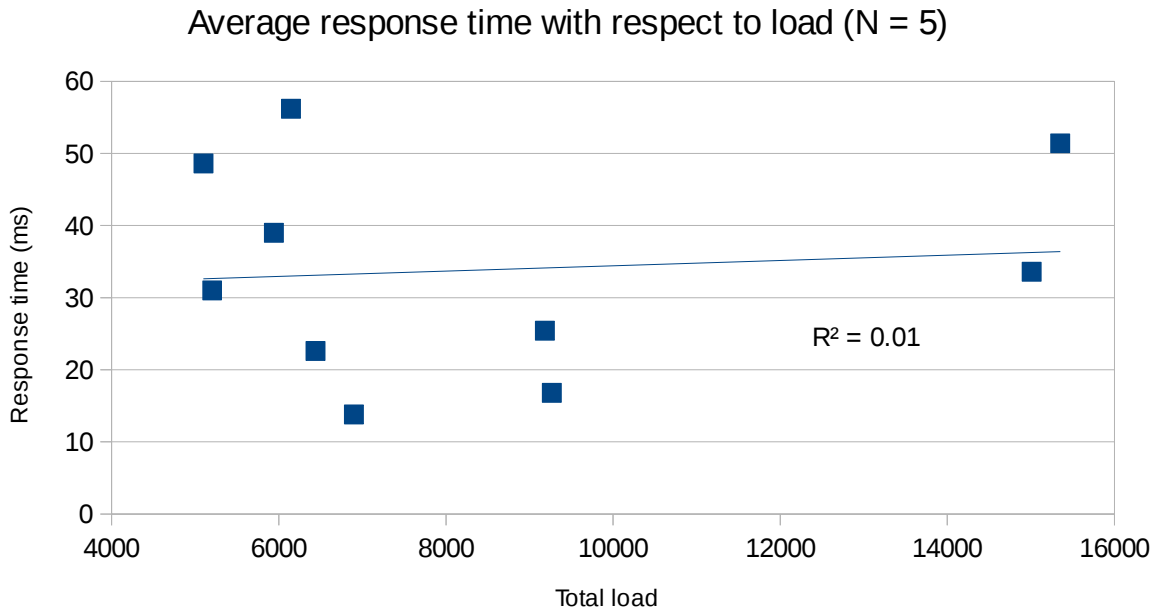
From these plots, we may observe that waiting and response times are both correlated positively with priority, waiting times more strongly than response times. That is, the lower priority a process is and the higher priority value it has, the longer it will wait for other processes to execute before it can utilize the CPU. This is natural by the design of the algorithm, as CFS schedules processes with minimum virtual runtime, which is inversely proportional to the weight of the process. Hence, the lower priority a process is, the higher its virtual runtime will be for the same actual runtime.

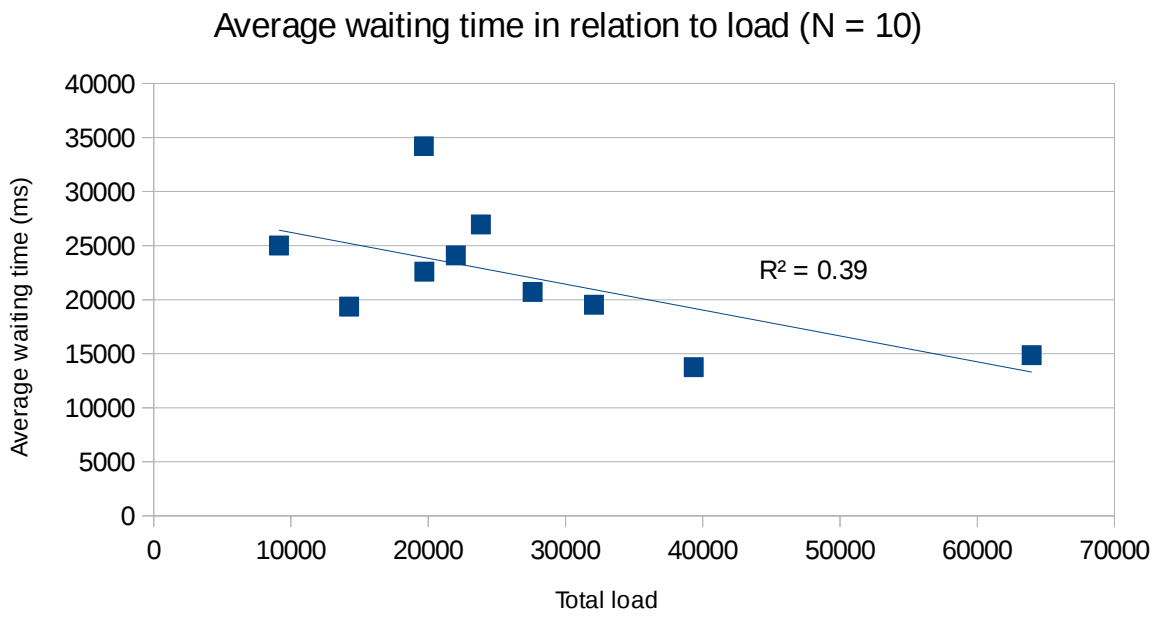
We may attribute the weaker correlation of response times with priority to the fact that CFS updates the virtual runtime of processes returning from an I/O burst based on the minimum virtual runtime of the runqueue, and independently of the priority of the process. If the priority of the returning process, or the total load of other processes, is high enough that its virtual runtime before the I/O burst falls below the minimum virtual runtime subtracted by the targeted latency, the virtual runtime of the process will be set to that fixed value. Hence, the time it takes for the process to be scheduled after returning from an I/O burst only depends on the priority of the process in certain cases.

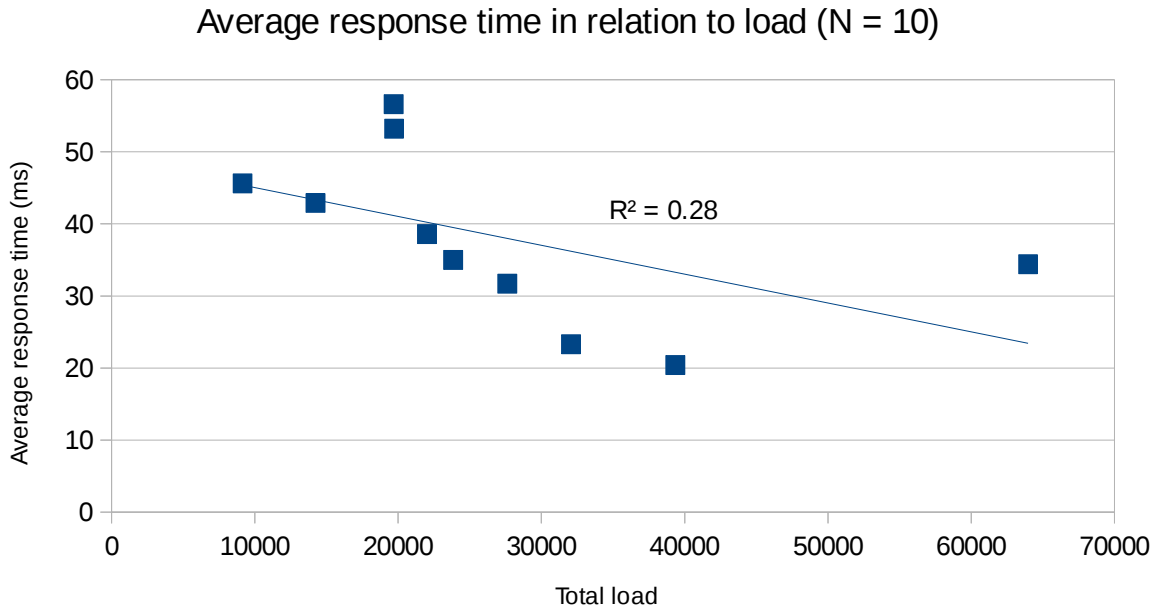
2. Average waiting and response times in relation to total load of workload file for fixed N

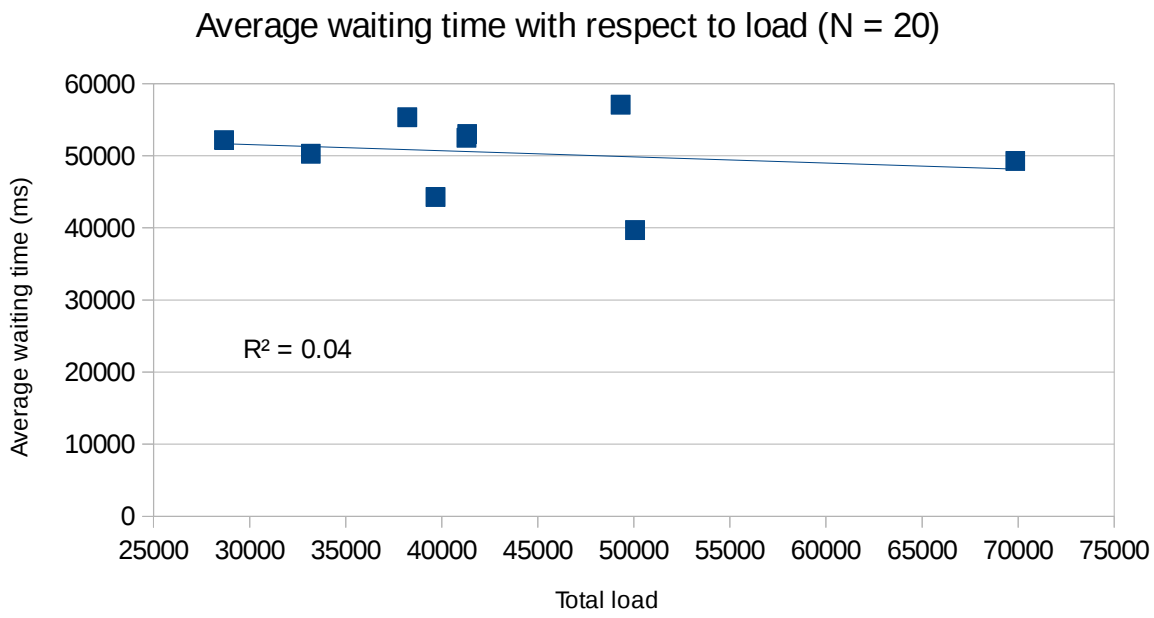
Additionally, for each N, 10 workload files were generated and simulated. The average waiting and response times of each workload file are plotted below in relation to the total load of each workload file, for fixed N.

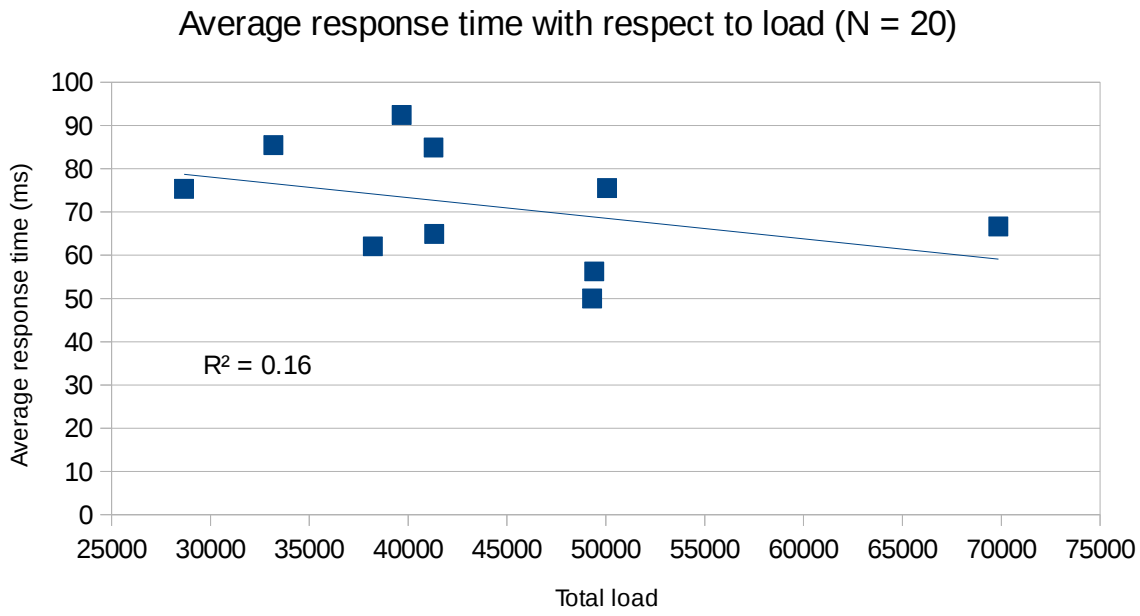





Average waiting time in relation to load (N = 10)
Average waiting time (ms)
40000
35000
30000
25000
20000
15000
10000
5000
0
0
10000
20000
30000
40000
50000
60000
70000
Total load
R² = 0.39


Average response time in relation to load (N = 10)
Average response time (ms)
60
50
40
30
20
10
0
0
10000
20000
30000
40000
50000
60000
70000
Total load
R² = 0.28





From these plots, we can observe a very weak negative correlation of both average waiting times and average response times with total load. This may be attributed to the fact that waiting and response times are dependent on the relative priority of a process compared to those of others, not necessarily the absolute priority.

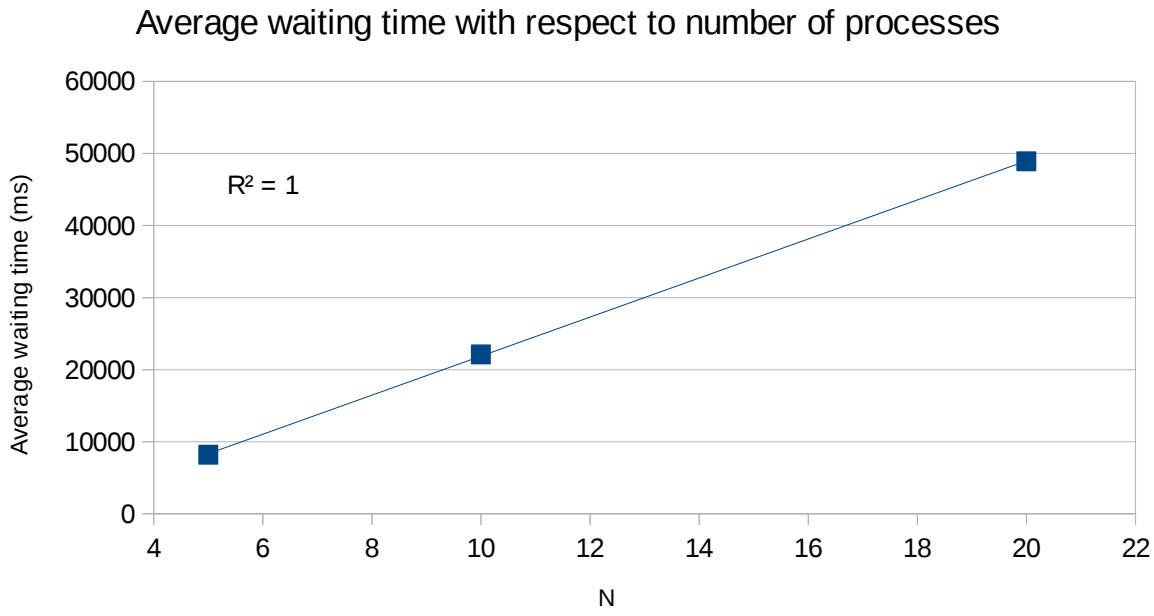
Since the weight of a process with priority n (centered at 20) is approximately equal to 1024*1.25^(20-n) [source: kernel/sched.c, lines 1345-1354], whether one process has a smaller vruntime than another depends only on the ratio of their runtimes and the difference of their priorities, not their exact values. If the priority of each process is increased or decreased by a constant value, the scheduler should work exactly the same. Similarly, the time slices allotted to each process depend only on the ratio of the weight of the process with the total load, which is again

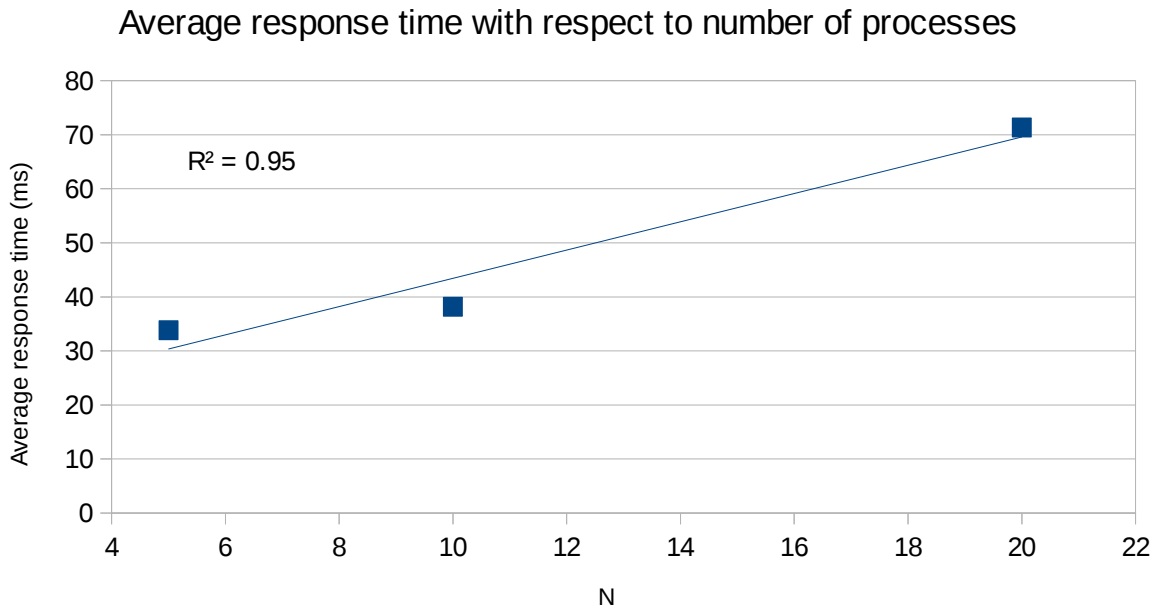
invariant under the translation of all priorities by a constant value; hence the time a process needs to wait in order to complete one CPU burst is invariant under the translation of priorities.

The slight correlation might be caused by idiosyncrasies in the workload files generated and changes in the load of the runqueue as processes enter I/O bursts, as well as variance in the weights of process. Nevertheless, waiting and response times should ideally be uncorrelated with the total load for fixed N, and the plots generated do not pose strong evidence against this.

3. Average waiting and response times in relation to N

The average waiting and response times of the 10 workload files for each N are lastly plotted below.





Here, we can observe that the average waiting and response times are both strongly dependent on the number of processes running. This is again expected, as the more processes are running, the longer a process will wait on average. Since the generated priorities, burst numbers and burst durations are independent and identically distributed random variables across different processes, the expected number of processes whose virtual runtime is less than that of the running process, averaged over all processes and scheduling times, will be about half the number of processes waiting.

We may also note that the average waiting times are much larger than the average runtime of each process, which is 10*400+9*100 = 4900 ms. In fact, the regression line for average waiting times follows 4900*(N-1)/2 very closely, suggesting that processes wait on average for half of the remaining processes to finish completely. This may again be attributed to the fact that the expected number of processes with less virtual runtime than a given process at a given moment in time will be (N-1)/2, and the process will wait for each of them to conclude their time slices before it can enter its.

The average response times, however, are much smaller than waiting times, firstly because they are only counted after a process returns from an I/O burst, and not every time it is preempted and has to wait for other processes to complete their time slices, and secondly because they are averaged over all bursts and not summed as waiting times are. The regression line for average response times is approximately f(N) = 5.6*(N-1)/2 + 20, which implies that processes wait a constant 20 ms on top of about 5.6 ms for each process with lower virtual runtime.

Still, because virtual runtimes are updated after returning from an I/O burst, when the process has high enough priority, it may be scheduled immediately (and in fact some high priority processes may have zero average response time), and when it has a low priority, it may wait for every other process to finish before being scheduled. This might be the reason for the constant term in the regression line, as some processes wait a fixed 0 ms and some might wait a fixed 100 ms for example. In fact, it can be seen from plots in section 1 that response times exhibit much more variation than waiting times, and hence the coefficients of determination are lower for response times. Waiting times are updated much more systematically than response times, which are only updated 10 times on average, and the sporadicity of each I/O burst also gives rise to greater variation in the system state at each arrival, and hence in each response time.