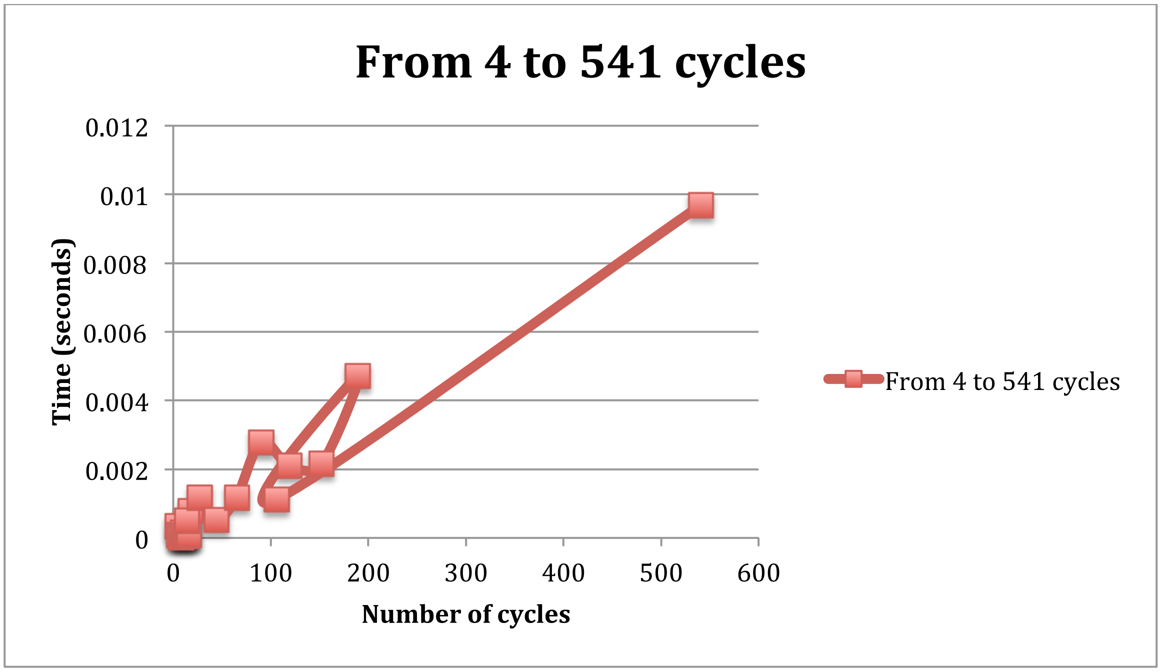
Wesley Moncrief
Jake Leland

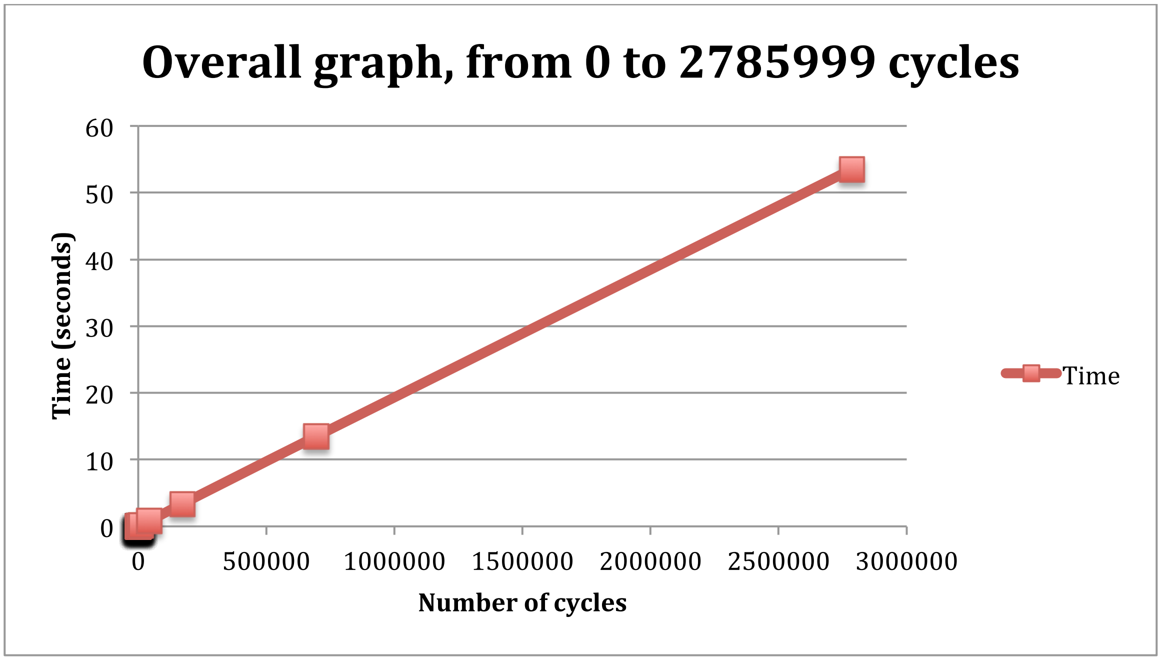
MP2 Report

In MP2, we implemented a buddy system for memory management. This data shows that there is a (imperfect) linear correlation between the number of allocation/free cycles and the total time. The fluctuations seen in data with fewer than 200 cycles can likely be attributed to random fluctuations in scheduler allocations of CPU time.

One example of a possible bottleneck in our system occurs in the my_free() function. Whenever we are preparing to merge blocks, we need to remove the two blocks from their free list, merge them together, and then add the merged block to the next tier of the list. Whenever we go to remove the half-blocks from their tier of the list, we know exactly where the first block is (it's at the front of the list), so it's easy to remove. However, we do not know the location of the buddy block in the linked list, so we have to increment through the list until we find it. Unfortunately, this is not an optimal method of finding the block. To fix this, we would need to make some changes to the way in which we keep track of the free blocks. For example, a better system of organizing/traversing the free list structure could provide a way for us to know where the buddy block will always be.

The performance of this system could be improved by implementing a less wasteful method of allocating data that requires slightly more than an allowable size, but significantly less than the next biggest allowable size. For example, if the basic block size is 32 and one wants to allocate 33 bytes of data, one must allocate 64 bytes (which wastes 31 bytes). This waste grows more significant as the size increases. A possible solution to this is Slab allocation.


From 4 to 541 cycles
Time (seconds)
0.012
0.01
0.008
0.006
0.004
0.002
0
0
100
200
300
400
500
600
Number of cycles
From 4 to 541 cycles


Overall graph, from 0 to 2785999 cycles
Time (seconds)
60
50
40
30
20
10
0
0
500000
1000000
1500000
2000000
2500000
3000000
Number of cycles
Time

The following data was used to obtain graph information.

```
Wesleys-MacBook-Pro:src wesmoncrief$ ./commLineTesting
    n = 1, m = 1
Result of ackerman(1, 1): 3
Time taken for computation : Seconds = 0.000341
 [sec = 0, musec = 341]
Number of allocate/free cycles: 4


    n = 1, m = 2
Result of ackerman(1, 2): 4
Time taken for computation : Seconds = 0.000074
 [sec = 0, musec = 74]
Number of allocate/free cycles: 6


    n = 1, m = 3
Result of ackerman(1, 3): 5
Time taken for computation : Seconds = 0.000012
 [sec = 0, musec = 12]
Number of allocate/free cycles: 8


    n = 1, m = 4
Result of ackerman(1, 4): 6
Time taken for computation : Seconds = 0.000146
 [sec = 0, musec = 146]
Number of allocate/free cycles: 10


    n = 1, m = 5
Result of ackerman(1, 5): 7
Time taken for computation : Seconds = 0.000071
 [sec = 0, musec = 71]
Number of allocate/free cycles: 12


    n = 1, m = 6
Result of ackerman(1, 6): 8
Time taken for computation : Seconds = 0.000059
 [sec = 0, musec = 59]
Number of allocate/free cycles: 14


    n = 1, m = 7
```

```
Result of ackerman(1, 7): 9
Time taken for computation : Seconds = 0.000089
 [sec = 0, musec = 89]
Number of allocate/free cycles: 16


     n = 1, m = 8
Result of ackerman(1, 8): 10
Time taken for computation : Seconds = 0.000776
 [sec = 0, musec = 776]
Number of allocate/free cycles: 18


     n = 2, m = 1
Result of ackerman(2, 1): 5
Time taken for computation : Seconds = 0.000497
 [sec = 0, musec = 497]
Number of allocate/free cycles: 14


     n = 2, m = 2
Result of ackerman(2, 2): 7
Time taken for computation : Seconds = 0.001166
 [sec = 0, musec = 1166]
Number of allocate/free cycles: 27


     n = 2, m = 3
Result of ackerman(2, 3): 9
Time taken for computation : Seconds = 0.000523
 [sec = 0, musec = 523]
Number of allocate/free cycles: 44


     n = 2, m = 4
Result of ackerman(2, 4): 11
Time taken for computation : Seconds = 0.001183
 [sec = 0, musec = 1183]
Number of allocate/free cycles: 65


     n = 2, m = 5
Result of ackerman(2, 5): 13
Time taken for computation : Seconds = 0.002785
 [sec = 0, musec = 2785]
Number of allocate/free cycles: 90
```

```
     n = 2, m = 6
Result of ackerman(2, 6): 15
Time taken for computation : Seconds = 0.002102
 [sec = 0, musec = 2102]
Number of allocate/free cycles: 119


     n = 2, m = 7
Result of ackerman(2, 7): 17
Time taken for computation : Seconds = 0.002153
 [sec = 0, musec = 2153]
Number of allocate/free cycles: 152


     n = 2, m = 8
Result of ackerman(2, 8): 19
Time taken for computation : Seconds = 0.004721
 [sec = 0, musec = 4721]
Number of allocate/free cycles: 189


     n = 3, m = 1
Result of ackerman(3, 1): 13
Time taken for computation : Seconds = 0.001124
 [sec = 0, musec = 1124]
Number of allocate/free cycles: 106


     n = 3, m = 2
Result of ackerman(3, 2): 29
Time taken for computation : Seconds = 0.009696
 [sec = 0, musec = 9696]
Number of allocate/free cycles: 541


     n = 3, m = 3
Result of ackerman(3, 3): 61
Time taken for computation : Seconds = 0.046724
 [sec = 0, musec = 46724]
Number of allocate/free cycles: 2432


     n = 3, m = 4
Result of ackerman(3, 4): 125
```

```
Time taken for computation : Seconds = 0.189261
 [sec = 0, musec = 189261]
Number of allocate/free cycles: 10307


     n = 3, m = 5
Result of ackerman(3, 5): 253
Time taken for computation : Seconds = 0.799054
 [sec = 0, musec = 799054]
Number of allocate/free cycles: 42438


     n = 3, m = 6
Result of ackerman(3, 6): 509
Time taken for computation : Seconds = 3.373090
 [sec = 3, musec = 373090]
Number of allocate/free cycles: 172233


     n = 3, m = 7
Result of ackerman(3, 7): 1021
Time taken for computation : Seconds = 13.469846
 [sec = 13, musec = 469846]
Number of allocate/free cycles: 693964


     n = 3, m = 8
Result of ackerman(3, 8): 2045
Time taken for computation : Seconds = 53.509865
 [sec = 53, musec = 509865]
Number of allocate/free cycles: 2785999
```