Malloc V. MyMalloc:

Experiment with large number of processes:

Data Analysis:

In the original experiment with 50 processes, we were able to observe that as the constraints were tightened, MyMalloc became faster than the library Malloc. We also noticed a trend towards a larger gap as the number of processes was increased. To further experiment on this issue, the total number of processes was changed from 50 to 100, then to 1000 to observer the changes in times. As we approached 100 processes, the times were in favor of Malloc, with it leading in the unconstrained tests. The interesting part was that when we added constraint to 10% available at any time, Malloc became quite a bit larger in overall time compared to MyMalloc. This same phenomenon was present in the second part of this experiment.

In the second part, we increased the process count to 1000 processes. As before, Malloc was faster than MyMalloc, but by much less than before. Then where thing really took a dramatic turn, was when we once again applied the 10% constraint, and this time the difference was very noticeable. The average time for malloc under these constraints, was 217,793.8 milliseconds, MyMalloc was 70,969.7 milliseconds. This is a difference of 206.88% between the two functions. This shows that our use of the tree for memory allocation greatly outperforms the malloc library function when placed under limited availability scenarios, which is a very common situation in day to day events in large systems. This outcome was definitely different from our original predictions.