Yao_Yishan_Program 1: Move-to-front Lists Brief Writeup

I took the initialized LinkedList/MTFList right out of the provided sample, i.e. 0 - 999 of size 1000, and below are the simulation output for both LinkedList and MTFList cases:

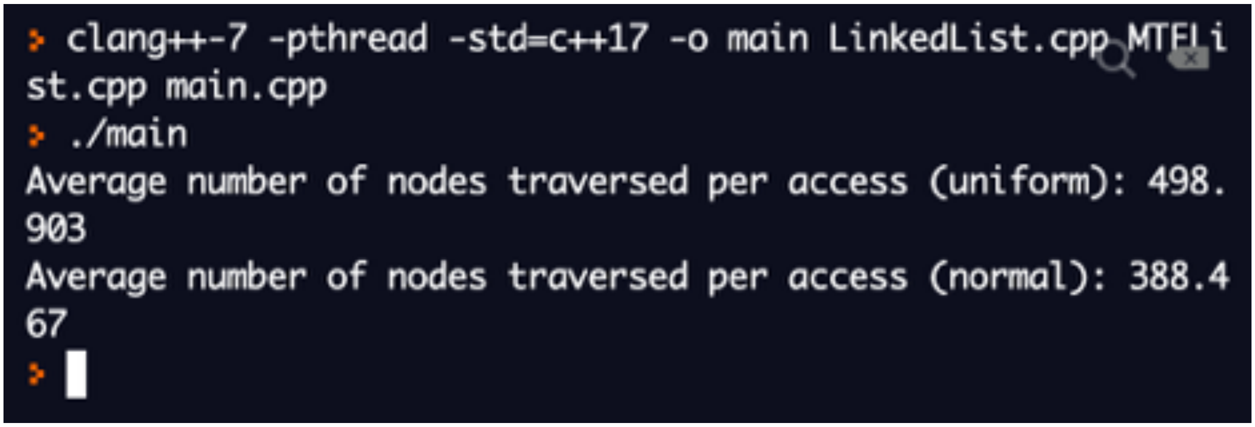
Uniform Distribution:
For Linked Lists uniform average: 500.778
For MTFList uniform average: 498.903

Under uniform distribution, since each access value has an equal probability of being drawn, therefore, the two implementations will have equivalent performance.

Normal Distribution
For Linked List normal average: 500.791
For MTFList normal average: 388.467

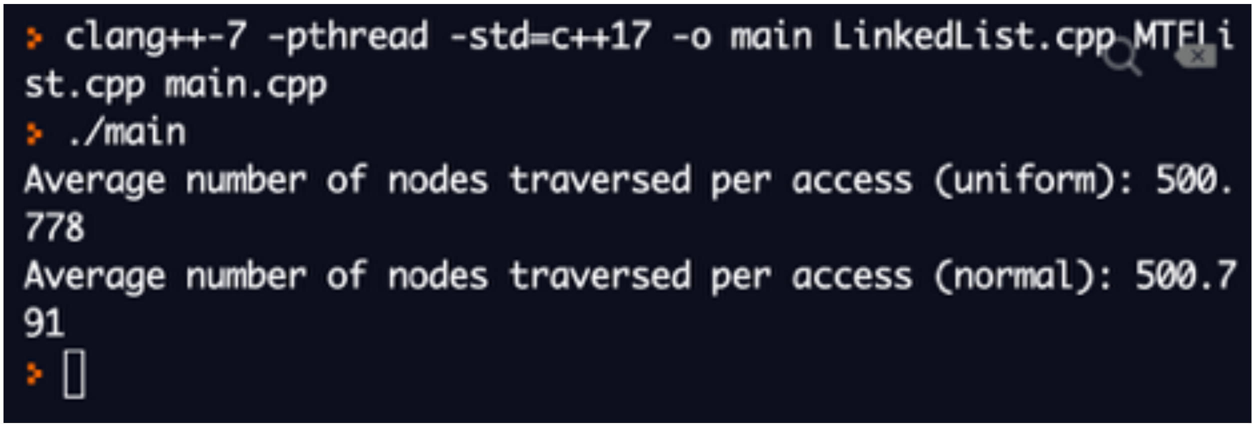
Under normal distribution, the random access value that is around the mean will have a higher probability of being drawn, hence, by moving these frequent occurrences to the front of the linked list will reduce the number of traverses when one is accessed for a second time. The simulation result backed up the assumption, where the average number of traverse in MTFList is significantly lower than that of a plain linked list.

MTFList uniform average and normal average:



```
> clang++-7 -pthread -std=c++17 -o main LinkedList.cpp MTFLi
st.cpp main.cpp
> ./main
Average number of nodes traversed per access (uniform): 498.
903
Average number of nodes traversed per access (normal): 388.4
67
> 
```

Linked Lists uniform average and normal average:



```
> clang++-7 -pthread -std=c++17 -o main LinkedList.cpp MTFLi
st.cpp main.cpp
> ./main
Average number of nodes traversed per access (uniform): 500.
778
Average number of nodes traversed per access (normal): 500.7
91
> 
```