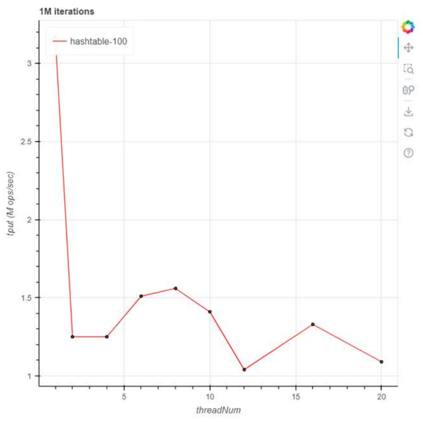
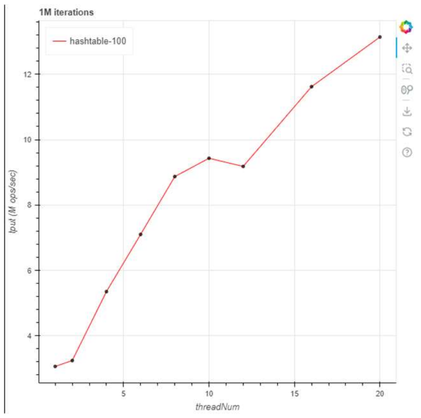
Tyler Kim
P2 Exp2b

There are a couple of things that were done to the `hashtable-biglock.c` skeleton file such that it functions correctly and scales well. The first implementation was to create an `H(k)` function that automatically takes a key to find which hashtable the key-value pair must go to using the equation, $H(key) = key \bmod (number\ of\ hashtables)$. In the `void *thread_func(void *thread_id)` function, I called `find_which_hashtable(int key)`, which gets the `H(k)` value for a given key, to extract the correct hash table index. I changed the arguments in the locking and unlocking mutex so that only the mutex that corresponds to the correct hash table index locks or unlocks. In addition, the key-value pairs are inserted into the correct hash table. Finally, in the `main(int argc, char **argv)` function, `numHashTables` was set to `100 * numThreads` to reduce contention for hash table resources.



Original Scalability Plot



After Improvements