

Govind Pillai
gopillai@ucsc.edu
6/6/2021

CSE 13s Spring 2021
Assignment 7: The Great Firewall of Santa Cruz
Lab Writeup

PURPOSE:

The purpose of this lab is to use a bloom filter, hash table, and linked lists in order to filter out words that are considered “bad”. The bloom filter is used as the first part of the filter which can most likely detect if a word is bad or not. However, it is possible for false positives to get through the bloom filter. If a word gets past the bloom filter, then a hash table with linked lists is used to confirm if the word was bad or was just a false positive that got past the bloom filter. Because the bloom filter size changes the amount of false positives that gets through, and the hash table size changes how many items are traversed through a linked list, that is what was graphed and studied for this writeup.

METHOD OF COLLECTING DATA:

There are four graphs included down below:

- The average seek length graphed against the hash table size
- The average seek length graphed against the hash table size w/ move-to-front
- The average seek length graphed against the bloom filter size
- The average seek length graphed against the bloom filter size w/move-to-front

Both variables (hash table and bloom filter size) were tested independently.

The average seek length is equal to the number of links over the number of seeks.

Results:

The first variable tested was the hash table size. What we find from the graph is that the average seek length drops as we increase the hash table size. Note that the average seek length is equal to the number of links over the number of seeks. The number of seeks does not change in this case because our input file still contains the same amount of words so we will still be doing the same amount of lookups. This means that our links value is dropping. This makes sense because As we increase our hash table size, it increases the amount of linked lists with the same amount of words. That means each linked list will be less populated. Therefore, when we traverse through a linked list, it is more likely that the word we are searching for is towards the beginning of the linked list.

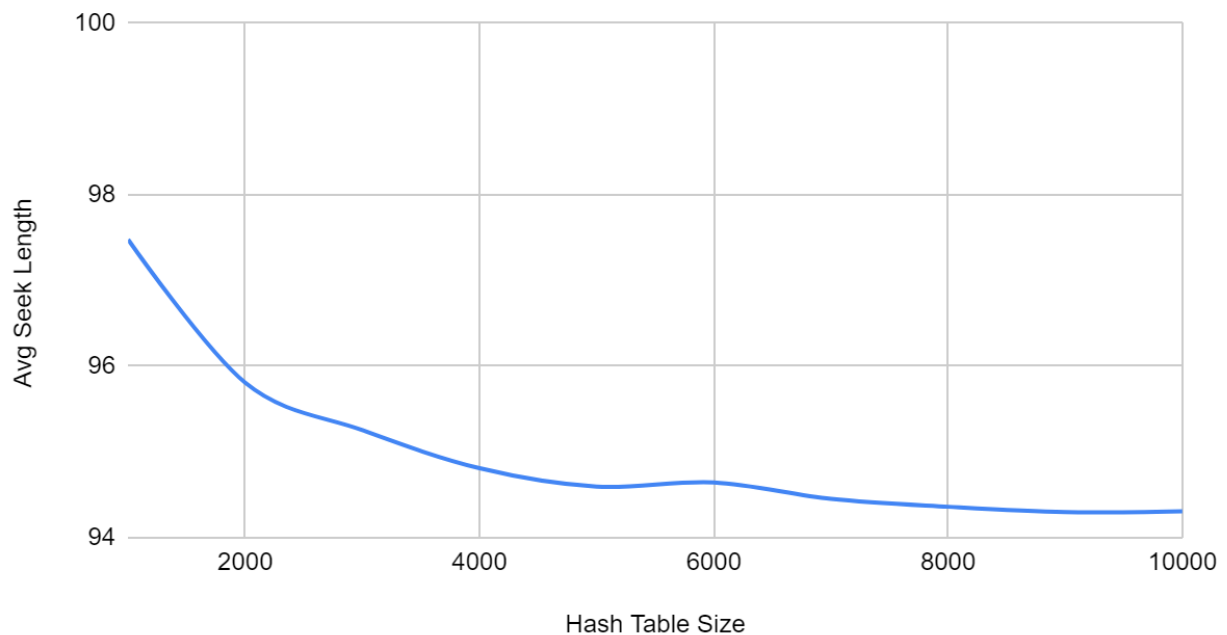
The next thing tested was the hash table size with move to front(mtf) enabled. That means every time we searched for a word, it would automatically move that word to the start of the linked list for future searches. As you can see, the graph follows the same pattern as the graph without mtf enabled. However, the value of the average seek length is drastically lower. This is because when we search for words many times, it is more likely that the element is already at the start of the linked list (because mtf is enabled). Therefore, we would have a smaller amount of links to traverse through.

The next variable tested was the bloom filter size. As the graph shows, as the bloom filter increases, so does the average seek length. Because the bloom filter size does not directly affect the amount of items in a linked list, we know the change in average seek length is not because of the links value. This must mean that the seeks value is getting lower. This makes sense because when our bloom filter size is small, we allow more false positives to get in. This means we would have to perform lookups for words that may not be bad. As we increase the bloom filter size, we decrease the amount of false positives and our lookups will mainly be used for bad words only.

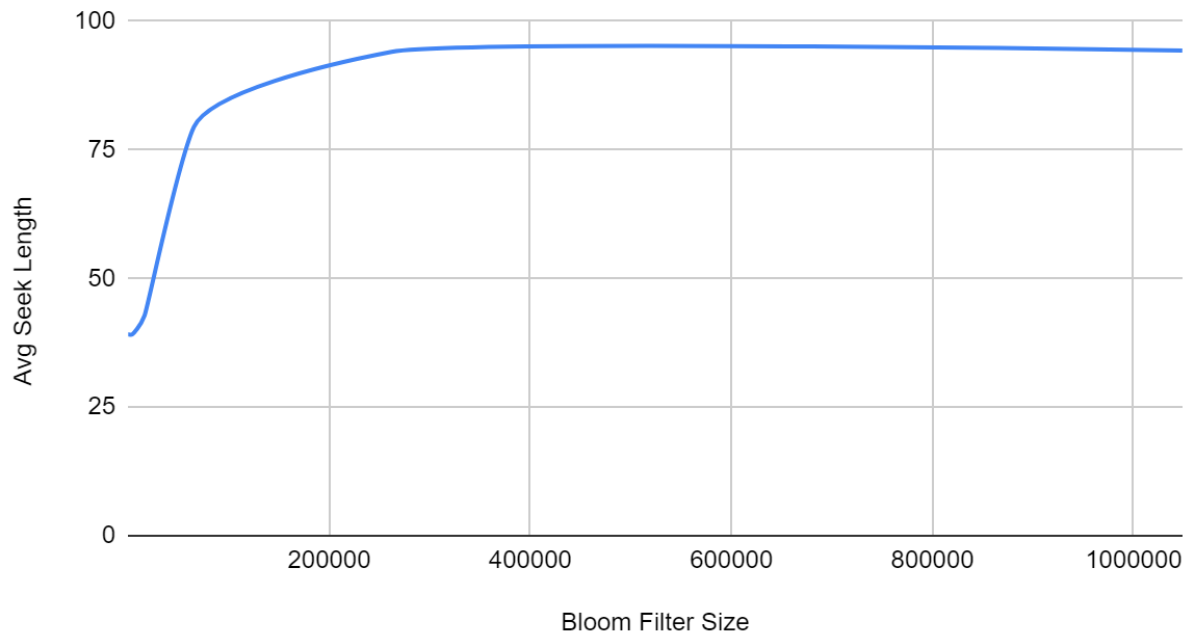
The next thing tested was the bloom filter size with mtf enabled. Once again, the graph follows the same pattern with much smaller seek length values. This is for the same reason why the seek length values were lower when changing the hash table size with mtf enabled. The number of links is decreasing when mtf is enabled because the items we search for are closer to the start of the linked list.

GRAPHS:

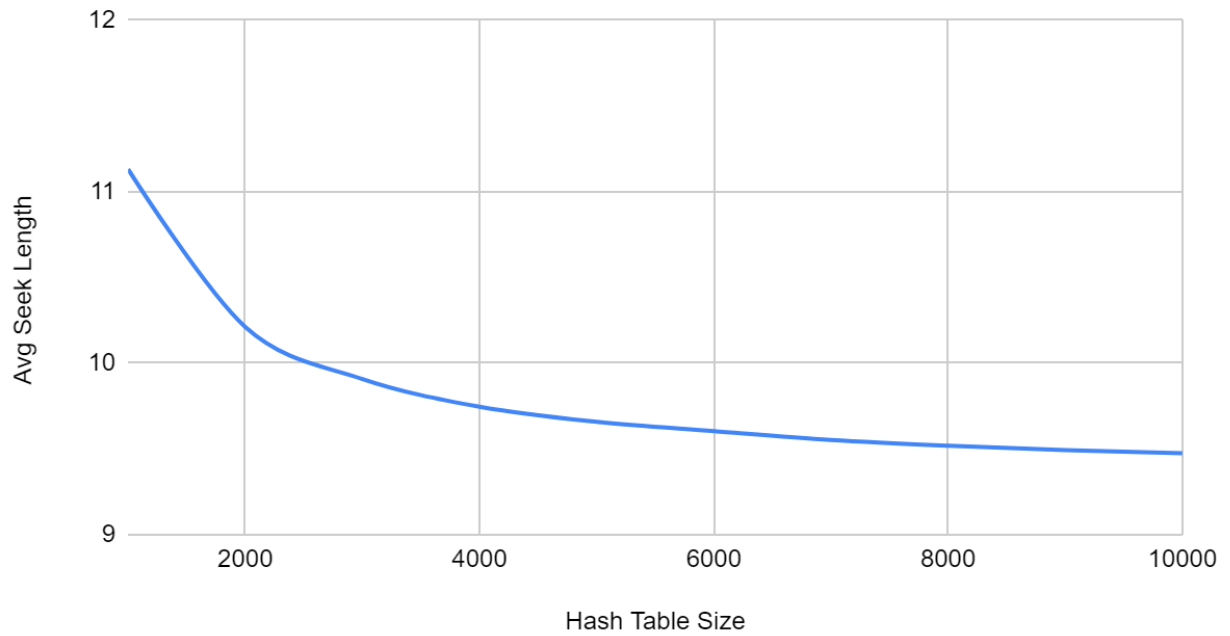
Avg Seek Length vs. Hash Table Size



Avg Seek Length vs. Bloom Filter Size



Avg Seek Length vs. Hash Table Size (mtf enabled)



Avg Seek Length vs. Bloom Filter Size (mtf enabled)

