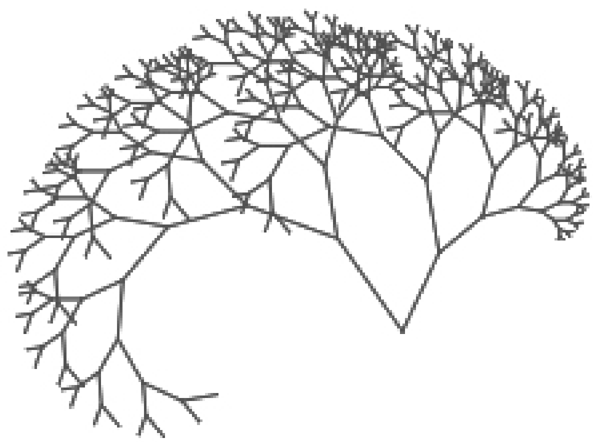
The Great Firewall of Santa Cruz: Write-up

Ralph Miller

December 5th 2021



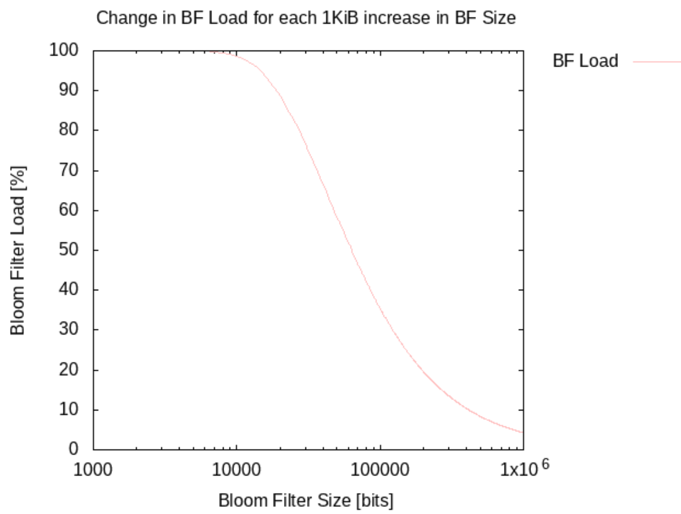
Contents

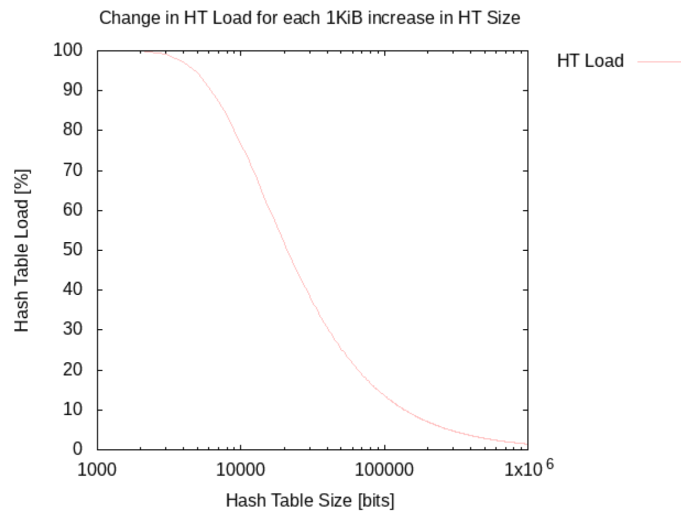
1 Introduction

In this write-up we will be analyzing the effect of Bloom filter and Hash table size using graphs created in GNU plot with 1KiB increments of size. The analysis will cover; BF and HT Load, Avg Branches Traversed, Number of lookups, and Avg BST height.

2 Analysis

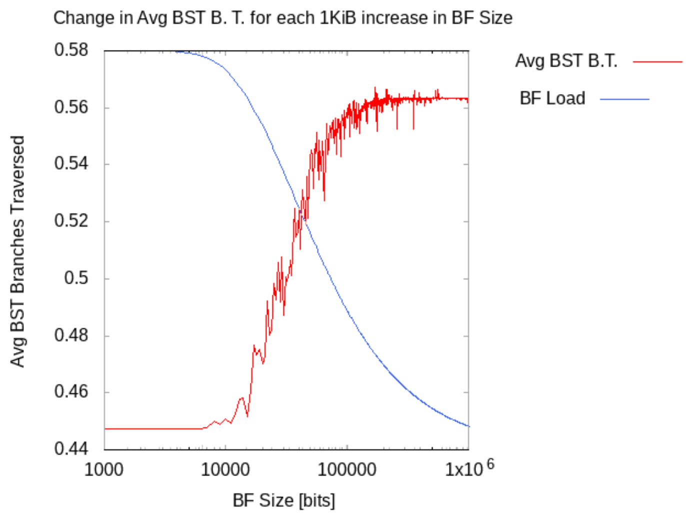
2.1 HT and BF Load Clarification

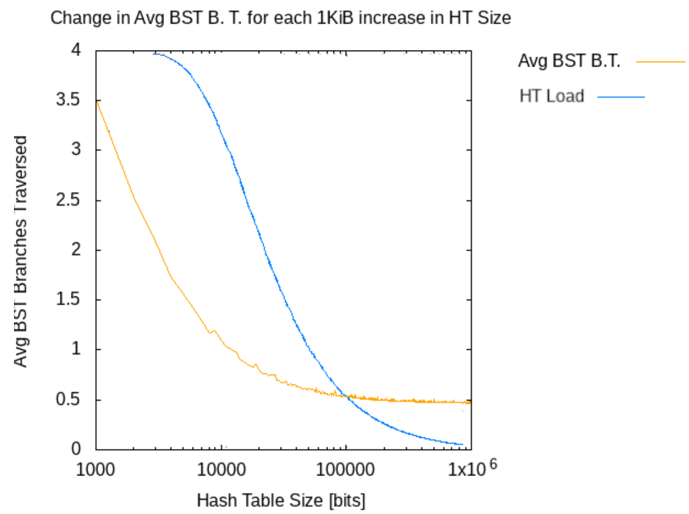




Above are the graphs that are used in all graphs that include a BF or HT Load. These graphs are super imposed onto another graph using GIMP Photo Editor. With that being said, I cannot guarantee pixel perfect accuracy, though I did my best to scale the load graphs to the graph it is super imposed on. Therefore, for future graphs the BF and HT load should be assumed as an approximate when interpreting the graphs.

2.2 Avg Branches Traversed

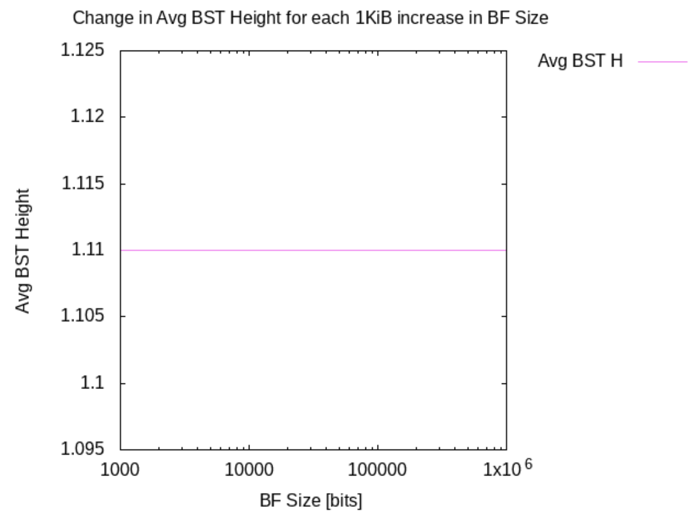


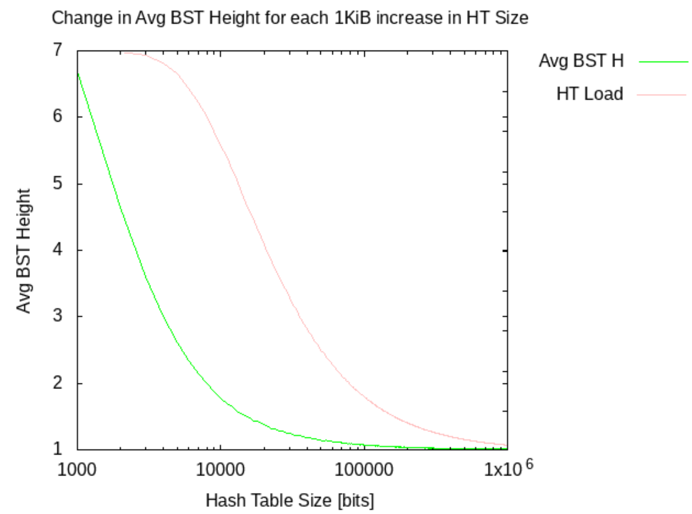


Above are graphs of the change in average binary search tree branches traversed in relationship to the Bloom filter size and hash table size. We can see that with each increase of the Bloom filter size and respective drop in Bloom filter load, the average branches traversed increases very slightly. We have a high rate of change around the 50,000 bit mark, which is very close to our default Bloom filter size of 65536.

Our total change in average branches traversed is much greater for increases in the hash table size. The greatest rate of change appears to happen between 1,000 and 10,000 bits. This happens as there are more binary search trees available for access in the hash table. So we have fewer hash collisions, and smaller binary search trees. We can see that as the hash stable size approaches 100,000 bits the average branches traversed flattens out to around the same value as the average branches traversed for a Bloom filter of that size.

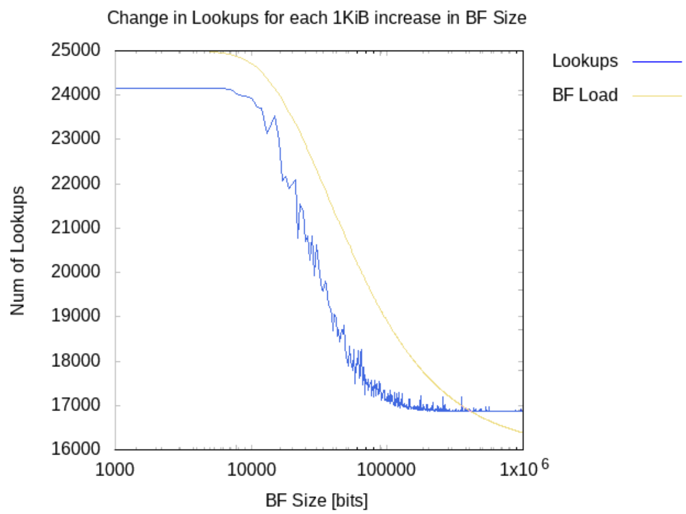
2.3 Avg BST Height

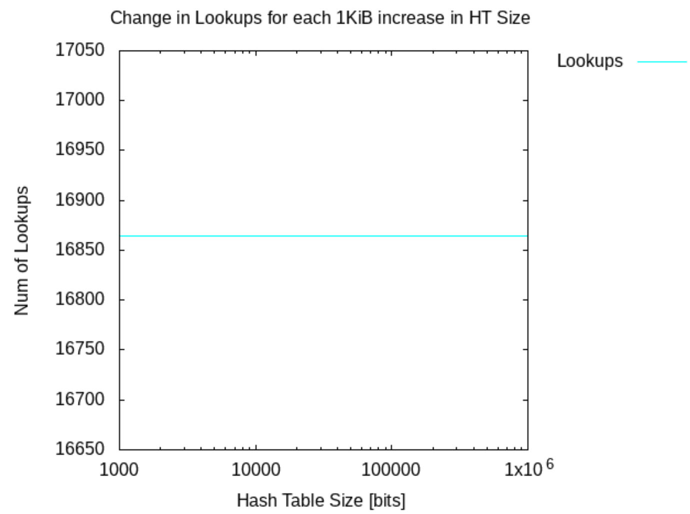




Above are graphs of the change in average binary search tree height in relationship to Bloom filter and hash table size. We can see that changing the size of the Bloom filter does not change the average binary search tree height as the Bloom filter is used to test if a binary search tree with a certain Node exists. What changes the average binary search tree height is the size of the hash table. We can see a rapid rate of change in average height from hash table sizes of 1,000 to 10,000. With the average height approaching 1 as the hash table size approaches 1,000,000. This is due to the fact that there must be at least 1 node in a binary search tree for a binary search tree to exist. The height also seems to be somewhat proportional to the Hash table load. With less load, more trees exist in the hash table, and average height would be smaller as nodes are evenly distributed.

2.4 Lookups





Above are the graphs of the change in number of lookups in relationship to Bloom filter and hash table size. We can see that the number of look ups is unresponsive to any hash table size. As a lookup only happens when the Bloom filter tests positive. We can see this relationship looking at the graph of lookups vs Bloom filter size. We have two extremes, 24000 look ups for a very small bloom filter and 17000 for a very large bloom filter. We have a high rate of change from 10,000 to 100,000 bits this is due to the increasingly fewer false positives due to hash collisions, as the bloom filter size increases, and its load decreases respectively. The likelihood of hash collisions decreases.