Com S 435 PA1-Bloom Filter

Jialin Xing

BloomFilterFNV:

Private method find_prime is used to find the smallest prime number that also larger than setSize * bitsPerElement, then use formula k=log(2)*filterSize/setSize to get the number of hash functions. To get hash value, it needs each char of the string h = h XOR s[i] and implement the formula. After that we need to convert from long 64 bit hash value into int, then we can store the hash value into the BitSet.

BloomFilterRan: randomly select a and b from {1,2,… ,p-1} to construct h(x)=ax+b. p is the prime number larger than setSize*bitsPerElement.

I import a list of words with about 3 thousand items and create four types of bloom filter. I change the value of each item randomly and use appear() function for each type of bloom filter. If there is one appear() return true which means one false positive happens, and use total number of false positive to get the false positive rate.

In the experiment, I see that bitsPerElement increases, the general false positive rate decreases. With same bitsPerElement, MultiMultiBloomFilter shows smallest false positive rates of all four (sometimes is 0 since data is not large enough). There is considerable difference between naiveBloomFilter and other three, since there is only one hash function being used. It is almost close to the theoretical prediction.

The time that not using BloomFilter is much longer than the one using Bloom Filter, especially when the data set is huge.