Probabilistic Near-Duplicate Detection Using Simhash

Arne Beer, MN 6489196

08.07.2019

1 Introduction

2 Conventional Hashing

Hashing is a technique, which is used to map data of an arbitrary size to a fingerprint with some fixed size. This procedure could be seen as a function $f(i) \rightarrow j$, which produces a value j from from any value i, where $j \in H$ and H is the set of values of the fixed length s with $s \in \mathbb{N}$. Well-known hash functions are, for instance, *md5* or *sha265*. These hashing functions are commonly used to check whether two files are absolutely identical or, for instance, to verify that a file has not been corrupted during transport. This is possible, since these hashing functions are designed to flip half of the output hash bits on average, if an input bit changes. Without this property it would be easier to change the input without the hash signature being modified. This would allow malicious third parties to, for instance, change code in a binary, without users being able to detect the change with the help of this hash and would require a full byte level comparison between the original and the copied file to verify its integrity.

If, on the other hand, one's goal is to find near duplicates, which are identical for the most part, but sometimes only differ by a few bits or bytes, this hashing approach immediately becomes useless, due to this property. Due to the need for a hashing algorithm, that creates a fingerprint based on the features and structure of the input data, *simhash* has been created.

2.1 Simhash

2.2

2.3 Achievements of Session Juggler

3 Impact in the scientific community

4 Relevance as of 2018