FSDS: Algorithms and figures

Vitor Hugo Galhardo Moia, Marco Aurélio Amaral Henriques

[1]School of Electrical and Computer Engineering (FEEC)
University of Campinas (UNICAMP)
Campinas, SP, Brasil 13083-852

vhgmoia,marco@dca.fee.unicamp.br

Obctober 19^{th}, 2017

1. Algorithm

```
input  : Number of buckets of the main table ($k$) and reference list objects
output: 0 (success)
Procedure PreparationPhase($refList$: list <string>) : int
    foreach $object$ in $refList$ do
        $digest$ = GenerateDigestTLSH($object$)
        $id$ = CreateId()
        $position$ = DistanceFunction($digest$, $referencePoint$)
        StoreDigestInMainTable($id$, $position$)
    end
    return 0
Procedure Main($refList$: list <string>, k: int) : int
    $size \leftarrow pow(2, k)$
    $mainTable \leftarrow$ CreateMainTable($size$)
    PreparationPhase($refList$)
    return 0
```

Algorithm 1: FSDS Preparation Phase

```
input  : Target system objects list ($targetList$), TLSH threshold ($t$), and
         radius ($r$) for searching similar objects in the hash table (main table)
output: List of similar pairs
Procedure OperationalPhase (targetList: list <string>, t: int, r: int) : list <string, string>
    foreach $object$ in $targetList$ do
        $digest$ = GenerateDigestTLSH($object$)
        $position$ = DistanceFunction($digest$, $referencePoint$)
        $candidatesList$ = findCandidatesOnMainTable($position$, $r$)
        foreach $candidate$ in $candidatesList$ do
            $score$ = CompareDigestsTLSH($digest$, $candidate$)
            if $score \leq t$ then
                $similarObjList$ = addToList($digest$, $candidate$)
            end
        end
    end
    return $similarObjList$
Procedure Main (targetList: list <string>, t: int, r: int) : list <string, string>
    $mainTable \leftarrow$ LoadMainTable()
    $list \leftarrow$ OperationalPhase (targetList, t, r)
    return $list$
```

Algorithm 2: FSDS Operational Phase

2. Figures

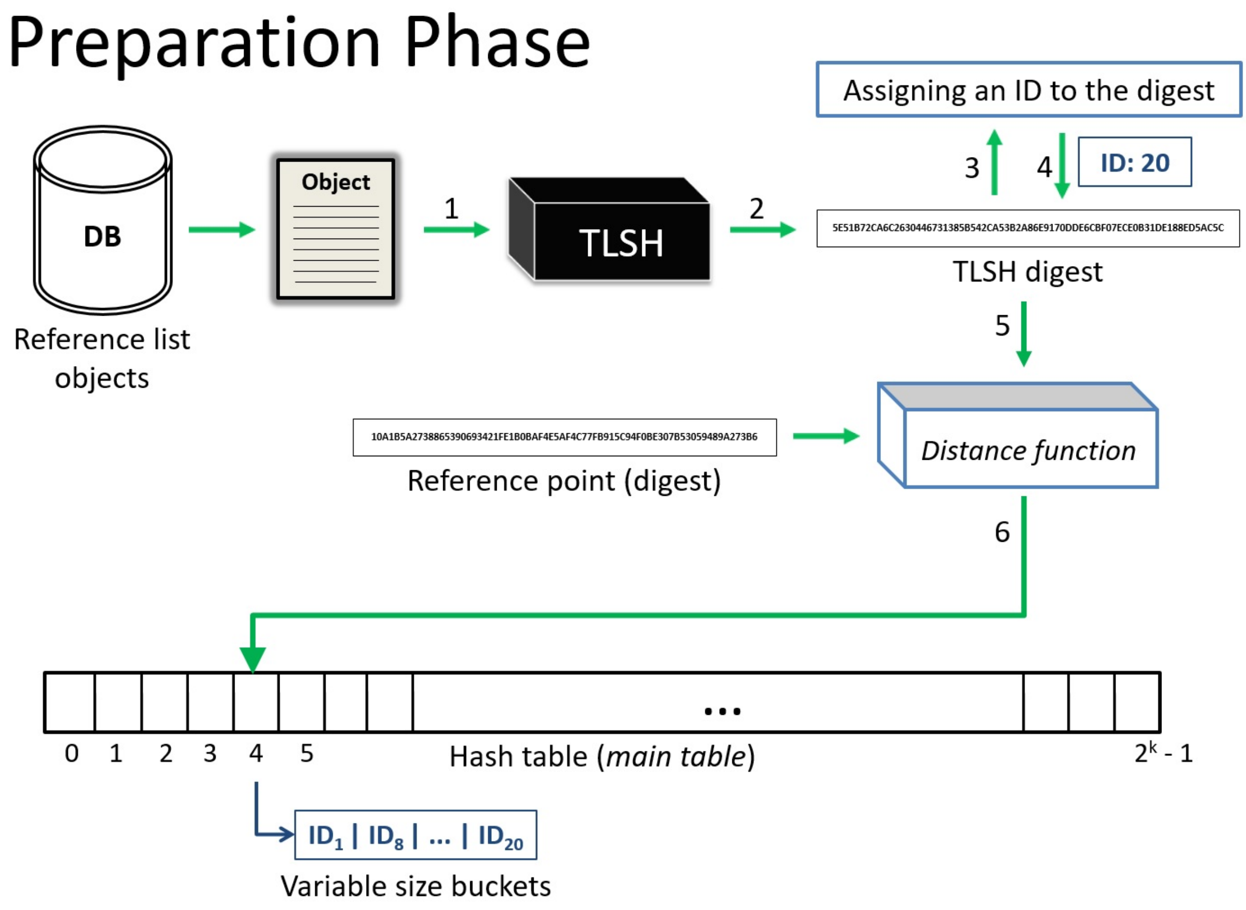
Preparation Phase
Assigning an ID to the digest
3
4
ID: 20
DB
Object
1
TLSH
2
5E51B72CA6C2630446731385B542CA53B2A86E9170DDE6CBF07ECE0B31DE188ED5AC5C
TLSH digest
Reference list objects
5
10A1B5A2738865390693421FE1B0BAF4E5AF4C77FB915C94F0BE307B53059489A273B6
Distance function
Reference point (digest)
6
...
0
1
2
3
4
5
Hash table (main table)
2^k - 1
ID_1 | ID_8 | ... | ID_{20}
Variable size buckets

Figure 1. Preparation phase

LEGEND:

1. Take an object.
2. Calculate its TLSH digest.
3. Create an ID to the digest.
4. Link the ID to the digest.
5. Apply the distance function on the digest and reference point.
6. Use the result distance to find the corresponding bucket in the main table to store the digest ID.

Operational Phase

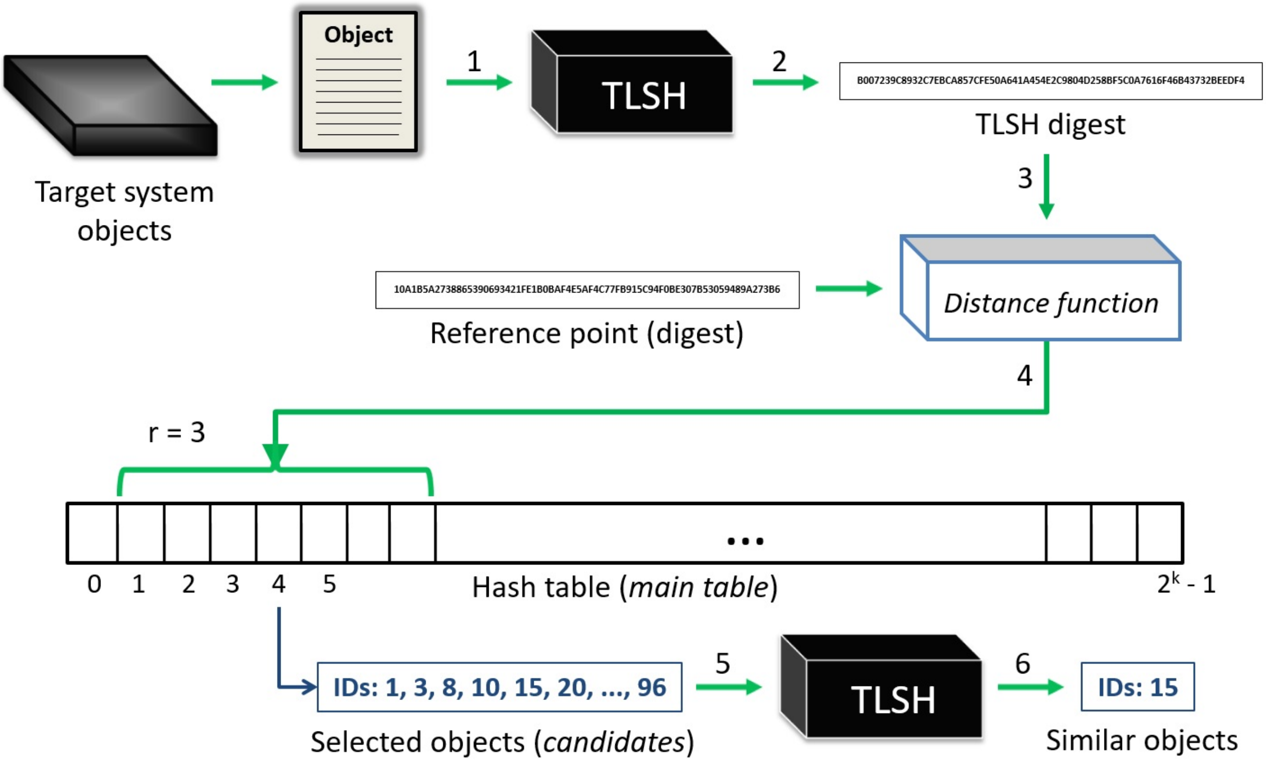
Target system objects
Object
1
TLSH
2
B007239C8932C7EBCA857CFE50A641A454E2C9804D258BF5C0A7616F46B43732BEEDF4
TLSH digest
3
10A1B5A2738865390693421FE1B0BAF4E5AF4C77FB915C94F0BE307B53059489A273B6
Reference point (digest)
Distance function
4
r = 3
...
0 1 2 3 4 5
Hash table (*main table*)
2^k - 1
IDs: 1, 3, 8, 10, 15, 20, ..., 96
Selected objects (*candidates*)
5
TLSH
6
IDs: 15
Similar objects

Figure 2. Operation phase

LEGEND:

1. Take an object.
2. Calculate its TLSH digest.
3. Apply the distance function on the digest and reference point.
4. Use the result distance to find the corresponding bucket in the main table and select all IDs in a radium *r*.
5. Perform a TLSH comparison of the queried object and all selected IDs in step 4.
6. All IDs below a certain threshold will be considered similar.