

multiple entities are involved. For example, two companies may have a need to exchange sets of documents stored in internal storage. Each company may use the same numbering system for identification, which may result in multiple documents having conflicting identifiers. As a result, either the documents must be stored in
5 separate tables and/or databases, or identifiers must be modified for one or both of the sets. In the former instance, it may be difficult to make adjustments to both tables/databases at once and may be inconvenient for the purposes of sorting and storage. In the latter instance, the two parties must both agree on the alternative identification with one or both parties modifying their previous identification system
10 accordingly, which can be highly inconvenient and potentially detrimental due to the prevailing use of the identifiers in the rest of their business. In addition, any future changes to identifiers, such as due to changing in business practices, or in an instance where a deal is struck with a third entity with whom there are conflicting identifiers, must be propagated to the other entity. As entities communicate and deal with more
15 and more entities, the conflicts may compound where changes to identifiers may occur at an unreasonable rate and negatively affect each entity's ability to do business.

Thus, there is a need for a technical solution to enable a universal distributed data storage where every data set, regardless of entity, type of data, method of storage, etc. can be combined without any collisions in identification, and
20 where the integrity is not affected by future additions of data to the data set.

SUMMARY

The present disclosure provide a method for generating a universal distributed data storage, comprising:

25 receiving, by a receiving device of a processing server, a data signal superimposed with a plurality of data values from two or more data sets to be combined into a single data storage without collisions;

30 generating, by a hashing module executed by a processor of the processing server, a key for each of the plurality of data values, wherein the respective key is generated via application of a collision-resistant hashing algorithm to the associated data value;

35 executing, by a querying module executed by the processor of the processing server, a first query on a key-value database of the processing server to store each of the plurality of data values and the associated generated key as a plurality of key-value pairs, where the generated key is a key in the key-value pair and where the data value is a value in the key-value pair;

40 receiving, by the receiving device of the processing server, a data signal superimposed with a plurality of additional key-value pairs, wherein each value of

the plurality of additional key-value pairs is not included in the plurality of data values;

executing, by the querying module of the processing server, a second query on the key-value database of the processing server to store each of the plurality of additional

5 key-value pairs;

receiving, by the receiving device of the processing server, a data signal superimposed with a rehash request, wherein the rehash request indicates a different hashing algorithm;

10 generating, by the hashing module of the processing server, a new key for each of the plurality of data values and each value included in the plurality of additional key-value pairs via application of the different hashing algorithm to the respective data value; and

15 executing, by the querying module of the processing server, a third query on the key-value database of the processing server to replace the generated key in each of the plurality of key-value pairs with the new key generated for the respective key-value pair, and to replace a key in each of the plurality of additional key-value pairs with the new key generated for the respective additional key-value pair without replacing the value in each of the plurality of

20 key-value pairs and the plurality of additional key-value pairs, wherein each key included in the plurality of additional key-value pairs is generated via application of the hashing algorithm to the associated value and is not included in the keys generated for each of the plurality of data values.

The present disclosure provides a system for generating a universal distributed data storage, comprising:

25 a key-value database of a processing server;

a receiving device of the processing server configured to receive a data signal superimposed with a plurality of data values from two or more data sets to be combined into a single data storage without collisions;

30 a hashing module executed by a processor of the processing server configured to generate a key for each of the plurality of data values, wherein the respective key is generated via application of a collision-resistant hashing algorithm to the associated data value; and

35 a querying module executed by the processor of the processing server configured to execute a first query on a database of the processing server to store each of the plurality of data values and the associated generated key as a plurality of key-value pairs, where the generated key is a key in the key-value pair and where the data value is a value in the key-value pair, wherein

40 the receiving device of the processing server is further configured to receive a data signal superimposed with a plurality of additional key-value pairs, wherein each value of the plurality of additional key-value pairs is not included in the plurality of data values,

the querying module of the processing server is further configured to execute a second query on the key-value database of the processing server to store each of the plurality of additional key-value pairs

45 the receiving device of the processing server is further configured to receive a data signal superimposed with a rehash request, wherein the rehash request indicates a different hashing algorithm,

the hashing module of the processing server is further configured to generate a new key for each of the plurality of data values and each value included in the plurality of additional key-value pairs via application of the different hashing algorithm to the respective data value,

the querying module of the processing server is further configured to execute a third query on the key-value database of the processing server to replace the generated key in each of the plurality of key-value pairs with the new key generated for the respective key-value pair, and to replace a key in each of the plurality of additional key-value pairs with the new key generated for the respective additional key-value pair without replacing the value in each of the plurality of key-value pairs and the plurality of additional key-value pairs, and each key included in the plurality of additional key-value pairs is generated via application of the hashing algorithm to the associated value and is not included in the keys generated for each of the plurality of data values.

The present disclosure is a method for generating a universal distributed data storage, comprising:

receiving, by a receiving device of a processing server, a data signal superimposed with a plurality of data values;

generating, by a hashing module of the processing server, a key for each of the plurality of data values, wherein the respective key is generated via application of a collision-resistant hashing algorithm to the associated data value;

executing, by a querying module of the processing server, a first query on a database of the processing server to store each of the plurality of data values and the associated generated key as a plurality of key-value pairs;

receiving, by the receiving device of the processing server, a data signal superimposed with a plurality of additional key-value pairs; generated via application of the collision-resistant hashing algorithm to the associated value;

executing, by the querying module of the processing server, a second query on the database of the processing server to store each of the plurality of additional key-value pairs, wherein each key included in the plurality of additional key-value pairs is generated via application of the collision-resistant hashing algorithm to the associated value;

receiving, by the receiving device of the processing server, a data signal superimposed with a rehash request, wherein the rehash request indicates a different collision-resistant hashing algorithm;

generating, by the hashing module of the processing server, a new key for each of the plurality of data values and each value included in the plurality of additional key-value pairs; and

executing, by the querying module of the processing server, a third query on the database of the processing server to replace the key in each of the plurality of key-value pairs and plurality of additional key-value pairs with the respective generated new key.

The present disclosure provides a description of systems and methods for generation of a universal distributed data storage. The use of collision-resistant hashing to develop identification values for disparate data sets may enable the

combination of data sets without collisions and may preserve integrity of the data sets and their identification following any future combination of additional data sets into the universal data storage.

A method for generating a universal distributed data storage includes:

- 5 receiving, by a receiving device of a processing server, a data signal superimposed with a plurality of data values; generating, by a hashing module of the processing server, a key for each of the plurality of data values, wherein the respective key is generated via application of a hashing algorithm to the associated data value; executing, by a querying module of the processing server, a first query on a database

10

WHAT IS CLAIMED IS:

1. A method for generating a universal distributed data storage, comprising:
 - 5 receiving, by a receiving device of a processing server, a data signal superimposed with a plurality of data values from two or more data sets to be combined into a single data storage without collisions;
 - 10 generating, by a hashing module executed by a processor of the processing server, a key for each of the plurality of data values, wherein the respective key is generated via application of a collision-resistant hashing algorithm to the associated data value;
 - 15 executing, by a querying module executed by the processor of the processing server, a first query on a key-value database of the processing server to store each of the plurality of data values and the associated generated key as a plurality of key-value pairs, where the generated key is a key in the key-value pair and where the data value is a value in the key-value pair;
 - 20 receiving, by the receiving device of the processing server, a data signal superimposed with a plurality of additional key-value pairs, wherein each value of the plurality of additional key-value pairs is not included in the plurality of data values;
 - 25 executing, by the querying module of the processing server, a second query on the key-value database of the processing server to store each of the plurality of additional key-value pairs;
 - 30 receiving, by the receiving device of the processing server, a data signal superimposed with a rehash request, wherein the rehash request indicates a different hashing algorithm;
 - 35 generating, by the hashing module of the processing server, a new key for each of the plurality of data values and each value included in the plurality of additional key-value pairs via application of the different hashing algorithm to the respective data value; and
 - 40 executing, by the querying module of the processing server, a third query on the key-value database of the processing server to replace the generated key in each of the plurality of key-value pairs with the new key generated for the respective key-value pair, and to replace a key in each of the plurality of additional key-value pairs with the new key generated for the respective additional key-value pair without replacing the value in each of the plurality of key-value pairs and the plurality of additional key-value pairs, wherein
 - 45 each key included in the plurality of additional key-value pairs is generated via application of the hashing algorithm to the associated value and is not included in the keys generated for each of the plurality of data values.
2. The method of claim 1, wherein the collision-resistant hashing algorithm is an SHA-256 algorithm.

3. The method of claim 1, further comprising:
executing, by the querying module of the processing server, a third query on the
database of the processing server to sort the plurality of key-value pairs and
5 plurality of additional key-value pairs based on the respective key.
4. The method of claim 1, wherein the different hashing algorithm is a collision-
resistant hashing algorithm.
- 10 5. The method of claim 4, wherein the different hashing algorithm is an SHA-256
algorithm.
6. The method of claim 1, further comprising:
executing, by the querying module of the processing server, a fourth query
15 on the database of the processing server to sort the plurality of key-value pairs and
plurality of additional key-value pairs based on the respective generated new key.
7. The method of claim 1, further comprising:
receiving, by the receiving device of the processing server, a data signal
20 superimposed with a value request, wherein the value request includes at least a
requesting key;
executing, by the querying module of the processing server, a third query
on the database of the processing server to identify a specific key-value pair
where the key corresponds to the requesting key; and
25 electronically transmitting, by a transmitting device of the processing
server, a data signal superimposed with at least the value included in the
identified specific key-value pair in response to the received value request.
8. The method of claim 1, wherein the collision-resistant hashing algorithm is
30 such that each generated key is unique with respect to all other generated keys.
9. A system for generating a universal distributed data storage, comprising:
a key-value database of a processing server;
a receiving device of the processing server configured to receive a data
35 signal superimposed with a plurality of data values from two or more data sets to
be combined into a single data storage without collisions;
a hashing module executed by a processor of the processing server
configured to generate a key for each of the plurality of data values, wherein the
respective key is generated via application of a collision-resistant hashing
40 algorithm to the associated data value; and
a querying module executed by the processor of the processing server
configured to execute a first query on a database of the processing server to
store each of the plurality of data values and the associated generated key as a
plurality of key-value pairs, where the generated key is a key in the key-value pair
45 and where the data value is a value in the key-value pair, wherein
the receiving device of the processing server is further configured to
receive a data signal superimposed with a plurality of additional key-value pairs,
wherein each value of the plurality of additional key-value pairs is not included in
the plurality of data values,

the querying module of the processing server is further configured to execute a second query on the key-value database of the processing server to store each of the plurality of additional key-value pairs

5 the receiving device of the processing server is further configured to receive a data signal superimposed with a rehash request, wherein the rehash request indicates a different hashing algorithm,

the hashing module of the processing server is further configured to generate a new key for each of the plurality of data values and each value included in the plurality of additional key-value pairs via application of the
10 different hashing algorithm to the respective data value,

the querying module of the processing server is further configured to execute a third query on the key-value database of the processing server to replace the generated key in each of the plurality of key-value pairs with the new key generated for the respective key-value pair, and to replace a key in each of
15 the plurality of additional key-value pairs with the new key generated for the respective additional key-value pair without replacing the value in each of the plurality of key-value pairs and the plurality of additional key-value pairs, and each key included in the plurality of additional key-value pairs is

generated via application of the hashing algorithm to the associated value and is not
20 included in the keys generated for each of the plurality of data values.

10. The system of claim 9, wherein the collision-resistant hashing algorithm is an SHA-256 algorithm.

25 11. The system of claim 9, wherein the querying module of the processing server is further configured to execute a third query on the database of the processing server to sort the plurality of key-value pairs and plurality of additional key-value pairs based on the respective key.

30 12. The system of claim 9, wherein the different hashing algorithm is a collision-resistant hashing algorithm.

35 13. The system of claim 12, wherein the different hashing algorithm is an SHA-256 algorithm.

14. The system of claim 9, wherein the querying module of the processing server is further configured to execute a fourth query on the database of the processing server to sort the plurality of key-value pairs and plurality of additional key-value pairs based on the respective generated new key.

40 15. The system of claim 9, further comprising:

a transmitting device of the processing server, wherein
the receiving device of the processing server is further configured to receive a data signal superimposed with a value request, wherein the value request
45 includes at least a requesting key,

the querying module of the processing server is further configured to execute a third query on the database of the processing server to identify a specific key-value pair where the key corresponds to the requesting key, and the

transmitting device of the processing server is configured to electronically transmit a data signal superimposed with at least the value included in the identified specific key-value pair in response to the received value request.

- 5 16. The system of claim 9, wherein the collision-resistant hashing algorithm is such that each generated key is unique with respect to all other generated keys.

17. A method for generating a universal distributed data storage, comprising:
receiving, by a receiving device of a processing server, a data signal
10 superimposed with a plurality of data values;
generating, by a hashing module of the processing server, a key for each of the plurality of data values, wherein the respective key is generated via application of a collision-resistant hashing algorithm to the associated data value;
executing, by a querying module of the processing server, a first query on
15 a database of the processing server to store each of the plurality of data values and the associated generated key as a plurality of key-value pairs;
receiving, by the receiving device of the processing server, a data signal superimposed with a plurality of additional key-value pairs; generated via application of the collision-resistant hashing algorithm to the associated value;
20 executing, by the querying module of the processing server, a second query on the database of the processing server to store each of the plurality of additional key-value pairs, wherein each key included in the plurality of additional key-value pairs is generated via application of the collision-resistant hashing algorithm to the associated value;
25 receiving, by the receiving device of the processing server, a data signal superimposed with a rehash request, wherein the rehash request indicates a different collision-resistant hashing algorithm;
generating, by the hashing module of the processing server, a new key for
30 each of the plurality of data values and each value included in the plurality of additional key-value pairs; and
executing, by the querying module of the processing server, a third query on the database of the processing server to replace the key in each of the plurality of key-value pairs and plurality of additional key-value pairs with the
35 respective generated new key.

18. The method of claim 17, wherein the collision-resistant hashing algorithm is an SHA-256 algorithm.

- 40 19. The method of claim 17, further comprising:
executing, by the querying module of the processing server, a third query on the database of the processing server to sort the plurality of key-value pairs and plurality of additional key-value pairs based on the respective key.

- 45 20. The method of claim 17, wherein the different collision-resistant hashing algorithm is an SHA-256 algorithm.