November 2022

DATA PARITY & RETENTION CHECKS IN DIFFERENT DATA CENTERS FOR FINDING DATA LEVEL MISMATCH

Rohini Ramesh Ms
Visa

TITLE OF THE INVENTION

DATA PARITY & RETENTION CHECKS IN DIFFERENT DATA CENTERS FOR FINDING DATA LEVEL MISMATCH

VISA

Rohini Ramesh

FIELD OF INVENTION:

The present invention generally relates to database systems and networking operations of multiple database centers with parallel data load pipelines. More particularly, but not specifically, the present invention relates to a method and system for automating the validation/ comparison of data and related objects from all the aspects between data centers to maintain parity as well as quality of data.

BACKGROUND:

The multi data center organizations often face the problem of keeping data at parity in all the data centers. For example, if there are two data centers datacenter 1 and datacenter 2, the number of tables, views, number of rows in each table, number of columns in each table, values in each column and data definition language of each table should exactly be the same.

Generally, the validation process involves one or more validation checks such as data type, range, format, consistency, uniqueness, etc. To accomplish data parity in different data centers with parallel data pipeline, the differences/mismatch need to be found proactively to take corrective action. In the earlier period, undertaking real time data migration involved rigorous manual effort, which could be extremely time-consuming besides expensive.

Accordingly, there were many challenges in the process such as validating a large volume of data like thousands of tables containing a huge amount of data in a particular time frame. In particular, validation becomes more challenging when each of those databases has its own structure. This makes the manual process of data parity and retention checks not feasible.

Therefore, automating the validation of data in multi - data centers to the mismatch in different data centers with parallel data pipelines becomes essential for real time data synchronization between those data centers to ensure accurate data as well as to facilitate switch over between data centers upon failure or other issues.

SUMMARY:

Various embodiments of the present invention provide a method and system to automate data parity and retention checks between data centers with parallel data load pipelines to ensure that the data within all of the databases is consistent across different sources and applications while a huge amount of information coming in from many sources. Data parity checks or data

synchronizations are required to be performed periodically to keep all the entries (tables and fields) matched between the data centers.

Accordingly, in an embodiment, the present invention discloses a method for automating data parity and retention check between data centers with parallel data load pipelines comprising, receiving by an automated validation system, one or more data entities for data parity and retention checks. The method comprises identifying, by the automated validation system, one or more key metrics to be used for data parity and retention check for data synchronization. Further the validation system also identifies one or more time dimension attribute within which the validation needs to be performed. Besides, the method comprises creating one or more text files based on one or more inputs received regarding one or more data entities. Upon identification of key metrics and time dimension attribute, the method creates one or more text files for validation regarding one or more data entities. Furthermore, the method generates dynamically a Structured Query Language (SQL) for each of the rows from the text files created by Unix script parsing through the rows as well as executes the generated scripts separately to capture the values for the corresponding datacenters. The method comprises, comparing the text files based on one or more metric values to get its difference and percentage of difference and updating each row by combining the captured results with the corresponding data entities for data parity and retention check.

According to a further aspect of the present invention, there is provided a computer implemented system for automating data parity and retention check between data centers with parallel data load pipelines configured to perform a method as defined above. The automated validation system comprises, an extraction module, for receiving one or more data entities for data parity as well as retention check and identifying one or more key metrics besides time dimension attributes within which the validation needs to be performed. Furthermore, the automated validation system configured to create one or more text files based on one or more inputs received regarding one or more data entities. Upon identification of key metrics and time dimension attribute, the system creates one or more text files for validation regarding one or more data entities. Furthermore, the system generates dynamically a Structured Query Language (SQL) for each of the rows from the text files created by Unix script parsing through the rows as well as executes the generated scripts separately to capture the values for the corresponding datacenters. The system further configured to compare the text files based on one or more metric values to get its difference and percentage of difference and updating each

row by combining the captured results with the corresponding data entities for data parity and retention check.

The foregoing summary is illustrative only and is not intended to be in any way limiting. In addition to the illustrative aspects, embodiments, and features described above, further aspects, embodiments, and features will become apparent by reference to the drawings and the following detailed description. For a better understanding of exemplary embodiments of the present invention, together with other and further features and advantages thereof, reference is made to the following description, taken in conjunction with the accompanying drawings.

BRIEF DESCRIPTION OF THE ACCOMPANYING DRAWINGS

The accompanying drawings, which are incorporated in and constitute a part of this disclosure, illustrate exemplary embodiments and together with the description, serve to explain the disclosed principles. In the figures, the left-most digit(s) of a reference number identifies the figure in which the reference number first appears. The embodiments of the disclosure itself, however, as well as a preferred mode of use, further objectives, and advantages thereof, will best be understood by reference to the following detailed description of an illustrative embodiment. Some embodiments of system and/or method in accordance with embodiments of the present subject matter are now described below, by way of example, and with reference to the accompanying figures.

Figure 1 is an illustration of system schematic of an exemplary validation system environment in multi datacentre networking in accordance with the present invention.

Figure 2 depicts a schematic diagram of an exemplary data parity and retention check process of validation system in multi datacentre networking environment in accordance with the present invention.

Figure 3 depicts a schematic block diagram of an exemplary automated validation system for data parity and retention check in a multi datacentre networking environment in accordance with the present invention.

Figure 4 depicts a schematic block diagram of an exemplary extraction module configured to create one or more text files based on one or more inputs received regarding one or more data entities in accordance with the present invention.

Figure 5 depicts a schematic block diagram of an exemplary transformation module for dynamically generating SQL scripts for each of the rows from the created one or more text files created in accordance with the present invention.

Figure 6 depicts a schematic block diagram of an exemplary comparison module for capturing the differences in the text files which compares each row by running queries in accordance with the present invention.

Figure 7 depicts a schematic block diagram of an exemplary retention module for data parity and retention check in a multi datacentre networking environment in accordance with the present invention.

Figure 8 depicts a flow diagram of an exemplary method for data parity and retention check in a multi datacentre networking environment in accordance with the present invention.

Figure 9 depicts a schematic block diagram of an exemplary computer system for data parity and retention check in a multi datacentre networking environment in accordance with the present invention.

The figures depict embodiments of the disclosure for purpose of illustration only. One skilled in the art will readily recognize from the following description that alternative embodiments of the structures and methods illustrated herein may be employed without departing from the principles of the disclosure herein.

It should be appreciated by those skilled in the art that any block diagrams herein represent conceptual views of illustrative systems embodying the principles of the present subject matter. Similarly, it will be appreciated that any flow charts, flow diagrams, state transition diagrams, pseudo code, and the like represent various processes which may be substantially represented in computer readable medium and executed by a computer or processor, whether or not such computer or processor is explicitly shown.

DESCRIPTION

In the present document, the word "exemplary" is used herein to mean "serving as an example, instance, or illustration." Any embodiment or implementation of the present subject matter described herein as "exemplary" is not necessarily to be construed as preferred or advantageous over other embodiments.

While the disclosure is susceptible to various modifications and alternative forms, specific embodiment thereof has been shown by way of example in the drawings and will be described in detail below. It should be understood, however that it is not intended to limit the disclosure to the particular forms disclosed, but on the contrary, the disclosure is to cover all modifications, equivalents, and alternative falling within the scope of the disclosure.

The terms “comprises”, “comprising”, or any other variations thereof, are intended to cover a non-exclusive inclusion, such that a setup, device, or method that comprises a list of components or steps does not include only those components or steps but may include other components or steps not expressly listed or inherent to such setup or device or method. In other words, one or more elements in a system or apparatus proceeded by “comprises… a” does not, without more constraints, preclude the existence of other elements or additional elements in the system or apparatus.

In the following detailed description of the embodiments of the disclosure, reference is made to the accompanying drawings that form a part hereof, and in which are shown by way of illustration specific embodiments in which the description may be practiced. These embodiments are described in sufficient detail to enable those skilled in the art to practice the disclosure, and it is to be understood that other embodiments may be utilized and that changes may be made without departing from the scope of the present disclosure. The following description is, therefore, not to be taken in a limiting sense.

Figure 1 illustrates an exemplary validation system environment (100) in multi datacentre networking in accordance with the present invention. The environment comprises one or more data centres 101 (101.a, 101.b, 101.c), with parallelized data load for which the data parity and retention checks are required to keep data at parity in all data centres. The data centres in the environment furthermore comprises a one or more databases 107 (107.a, 107.b, 107.c), also referred to as storage memory or main memory. The one or more databases is collectively referred as data centres (101) in the present disclosure. The data centres 101 may provide memory space, i.e., memory sections, assigned for use by hardware, firmware, and software components comprised by the environment with parallel data load (109). The database 107 may be used by hardware and firmware of environment as well as by software, e.g., hypervisors, host/guest operating systems, application programs etc. An automated validation system may be implemented in a computing system (103) is configured to automate data parity and retention checks between data centers with parallel data load pipelines to ensure that the

data within all of the databases is consistent across different sources and applications while a huge amount of information coming in from many sources.

Figure 2 depicts a schematic diagram of an exemplary data parity and retention check process of validation system in multi datacentre networking environment (200) in accordance with the present invention. The environment comprises a computer system (201) configured to perform data parity and retention check for the required data entities on periodic basis. For example, the data centres (203.x) and (203.y) comprises databases (205.x) & (205.y) with parallelized data load for which the data parity and retention checks are required to keep data at parity (should be exactly the same) in various data entities (207.x) & (207.y) such as but not limited to the number of tables, views, number of rows in each table, number of columns in each table, values in each column and data definition language of each tables. Thus, the present invention provides synchronized view(s) (209.x) & (209.y) of data in multi data centre networking environment.

Figure 3 illustrates a schematic block diagram of an exemplary automated validation system for data parity and retention check in a multi datacentre networking environment (300) in accordance with the present invention. As illustrated in Fig.3, the automated validation system (303) includes one or more blocks for automating the process of keeping data at parity in all data centres. The system (303) comprises, an extraction module (305), Transformation Module (307), Comparison Module (309), and Retention Module (311) for providing data synchronization with parallel load pipeline. Upon receiving a list of data entities (301), the system provides data parity and retention to ensure that the data within all of the databases is consistent across different sources (315) and applications while a huge amount of information coming in from many sources (313). However, various versions of the modules involved for automating the process of keeping data at parity in all data centres, may be equally configured or adapted to implement embodiments for various other types of database systems. Therefore, the following examples are not intended to be limited as to various other types or formats of database systems.

In some embodiments, multiple users may access a distributed datacentres to obtain various services. The multiple users may include various applications and/or data warehouse service audience. For example, the captured difference in various data entries may be communicated to those expected audience such as, (but not limited to) sending mail with corresponding difference and percentage calculation in body of the mail.

Figure 4 depicts a schematic block diagram (400) of an exemplary extraction module configured to create one or more text files based on one or more inputs received regarding one or more data entities in accordance with the present invention.

In yet another embodiment, the extraction module (407) comprises key metrics identification module (409.x) for identifying one or more key metrics as well as time dimension attribute identifying module (411.x) based upon the received list (401) of data entities. Besides, the extraction module (407) further configured to create text files (415.x) & (415.y) with inputs received regarding data entities.

For an example, the text file is created for two base table as shown in the table 1.

Table: 1 – Creation of Text File for two base table

View Name	Base Table	sum(AMT Fact)	sum(CNT Fact)	Month
VIEW1.VEDF_AUTH_ACQR	TABLE1.TEDF_AUTH_ACQR	AUTH_TRAN_AMT	AUTH_TRAN_CNT	CPD_MNTH_ID
VIEW2.VEDF_AUTH_ACQR	TABLE2.TEDF_AUTH_AIM_NSPK	AUTH_TRAN_AMT	AUTH_TRAN_CNT	CPD_MNTH_ID

In an embodiment, the list of data entities for validation may belong to different data centres. The data entities (405) may include, but not limited to tables, records, etc. Further, the key metrics may include, but not limited to Amount and Count. Furthermore, the time frame attributes may include, but limited to month and date field.

Figure 5 illustrates a schematic block diagram (500) of an exemplary transformation module for dynamically generating SQL scripts for each of the rows from the created one or more text files created in accordance with the present invention.

In an embodiment, the transformation module (503) comprises Transformation Logic module (505.x) as well as SQL generator module (507.x) for dynamically generating SQL scripts (509.x) for each of the rows from the created one or more text files (501.x).

For example, the dynamically generating the SQL for each row from the generated text file as shown below:

```
db2 -x $'select
(VIEW2.VEDF_AUTH_ACQR'),  (TABLE2.TEDF_AUTH_AIM_NSPK'),'AUTH_TRAN_
AMT', sum(cast (AUTH_TRAN_AMT as bigint)), 'AUTH_TRAN_CNT', sum(cast
```

```
(AUTH_TRAN_CNT as bigint)) from TABLE2.TEDF_AUTH_AIM_NSPK  where
CPD_MNTH_ID=201903 group by  CPD_MNTH_ID  with ur'
```

In an embodiment, the transformation module may include the process of Unix script parse through the rows from the text file and dynamically creates a SQL. Furthermore, the SQL runs the sum on facts only for the last month, to ensure the data parity is checked for the last/recent one month, as the process of validation continues every month.

Figure 6 depicts a schematic block diagram (600) of an exemplary comparison module for capturing the differences in the text files which compares each row by running queries in accordance with the present invention.

In yet another embodiment, the comparison module (603) comprises SQL execution logic module (605.x). The method further provides a method for compare each row from different data centres, by running queries (607). Thus, captures the difference values in text files (609) created from the received list of data entities and provides difference analysis results (611) for further checking and retention process.

Figure 7 illustrates a schematic block diagram (700) of an exemplary retention module for data parity and retention check in a multi datacentre networking environment in accordance with the present invention.

In yet another embodiment, the retention module (703) comprises SQL execution logic module (705.x) configured to execute the generated SQL in the corresponding database to capture the values in two different text files. Upon the execution of queries in various databases, the module further combines the text files with the captured results to update (707) the databases correspondingly. The executed results regarding differences and percentage of differences (713) during the data parity and retention may be communicated to the expected audiences (715). For example, the scripts (701) executed at the datacentres (705.x) and (705.y) to keep the data exactly same in all of the data centres.

```
For E.g.: IN 705.x
View_Name          Table_Name          Amt_Metric   Value              Cnt_Metric   Value   Max
_Mnth Min_Mnth
VIEW1.VEDF_AUTH_ACQR     TABLE1.TEDF_AUTH_ACQR     AUTH_TRAN_AMT  3498808176      A
UTH_TRAN_CNT  621589    201604  200912
VIEW2.VEDF_AUTH_ACQR     TABLE2.TEDF_AUTH_AIM_NSPK   AUTH_TRAN_AMT  162257        A
UTH_TRAN_CNT  864      201903  201511
```

For E.g.: IN 705.y

View_Name Table_Name Amt_Metric Value Cnt_Metric Value Max _Mnth Min_Mnth

VIEW1.VEDF_AUTH_ACQR TABLE1.TEDF_AUTH_ACQR AUTH_TRAN_AMT 398808176 AU TH_TRAN_CNT 62189 201604 200912

VIEW2.VEDF_AUTH_ACQR TABLE2.TEDF_AUTH_AIM_NSPK AUTH_TRAN_AMT 12257 AU TH_TRAN_CNT 86 201903 201511

Figure 8 depicts a flow diagram of an exemplary method for data parity and retention check in a multi datacentre networking environment in accordance with the present invention. At step 801, the method includes, receiving, by an automated validation system, one or more data entities for data parity and retention checks and identifying, by the automated validation system, one or more key metrics to be used for parity check at step 803 as well as identifying, by the automated validation system, one or more time dimension attribute within which the validation needs to be performed at step 805.

Further, the method includes, creating, by the automated validation system, one or more text files based on one or more inputs received regarding one or more data entities at step 807 and dynamically creating, by the automated validation system, a SQL for each of the rows from the text files created by Unix script parsing through the rows at step 809.

Furthermore, the method includes, executing, by the automated validation system, the generated scripts separately to capture the values for the corresponding datacentres at step 811. The method also performs comparison of text files based on one or more metric values to get its difference and percentage of difference and updates each row by combining text files, at step 813. Also, the method communicates the updated data entities based on the captured execution results to the expected audience, at step 815. For example,

705.x	**705.x**	**705.x**	**705.x**	**705.x - 705.y (COUNT)**	**705.x - 705.y /705.x % (COUNT)**	**705.x**	**705.x -705.y (COUNT)**	**705.x - 705.y /705.x % (AMOUNT)**
View Name	**Table Name**	**CPD MONTH**	**Tran COUNT**	**Diff COUNT**	**Percentage COUNT**	**Tran AMOUNT**	**Diff COUNT**	**Percentage AMOUNT**

VIEW1.VEDF_AUTH_ACQR	TABLE1.TEDF_AUTH_ACQR	200912	3498808176	3100000000	88.6016	621589	559400	89.9952
VIEW2.VEDF_AUTH_ACQR	TABLE2.TEDF_AUTH_AIM_NSPK	201511	162257	150000	92.4459	864	778	90.0463

Figure 9 illustrates a schematic block diagram of an exemplary computer system (901) for data parity and retention check in a multi datacentre networking environment in accordance with the present invention. The computer system comprises a plurality of processors (903.a, 903.b, 903.c, etc.), I/O interface with parallel load (905), Memory (907), Network Interface (909), and I/O devices (911).

The terms "an embodiment", "embodiment", "embodiments", "the embodiment", "the embodiments", "one or more embodiments", "some embodiments", and "one embodiment" mean "one or more (but not all) embodiments of the invention(s)" unless expressly specified otherwise.

The terms "including", "comprising", “having” and variations thereof mean "including but not limited to", unless expressly specified otherwise.

The enumerated listing of items does not imply that any or all of the items are mutually exclusive, unless expressly specified otherwise. The terms "a", "an" and "the" mean "one or more", unless expressly specified otherwise.

A description of an embodiment with several components in communication with each other does not imply that all such components are required. On the contrary a variety of optional components are described to illustrate the wide variety of possible embodiments of the invention.

When a single device or article is described herein, it will be readily apparent that more than one device/article (whether or not they cooperate) may be used in place of a single device/article. Similarly, where more than one device or article is described herein (whether or not they cooperate), it will be readily apparent that a single device/article may be used in place of the more than one device or article, or a different number of devices/articles may be used instead of the shown number of devices or programs. The functionality and/or the features of a device may be alternatively embodied by one or more other devices which are not explicitly described as having such functionality/features. Thus, other embodiments of the invention need not include the device itself.

Finally, the language used in the specification has been principally selected for readability and instructional purposes, and it may not have been selected to delineate or circumscribe the inventive subject matter. It is therefore intended that the scope of the invention be limited not by this detailed description, but rather by any claims that issue on an application based here on.

While various aspects and embodiments have been disclosed herein, other aspects and embodiments will be apparent to those skilled in the art. The various aspects and embodiments disclosed herein are for purposes of illustration and are not intended to be limiting, with the true scope and spirit being indicated by the following claims.

Reference Number	Description
103	Automated validation system
105	Network interface
107	Plurality of Data centers
109	users
201	computer system
203	Plurality of Data centers
205	Plurality of Databases
207	Plurality of Data entities
209	synchronized view(s)
301	Data entries
303	Automated validation system
305	Extraction module
307	Transformation module
309	Comparison module
311	Retention module
313	Sources
315	View(s)
507	SQL generator
901	Computer system
903	Plurality of processors
905	I/O interface

907	Memory
909	Network interface

ABSTRACT

Various embodiments of the present invention provide a method and system to automate data parity and retention checks between data centers with parallel data load pipelines. The method comprises, receiving one or more data entities; identifying one or more key metrics to be used; identifies one or more time dimension attribute within which the validation needs to be performed; creating one or more text files based on one or more inputs received; identification of key metrics and time dimension attribute; generates dynamically a Structured Query Language (SQL) for each of the rows from the text files as well as executes the generated scripts separately to capture the values for the corresponding datacenters; and compares the text files based on one or more metric values to get its difference and percentage of difference and updates the captured results.

[Figure 1]

Figure 1

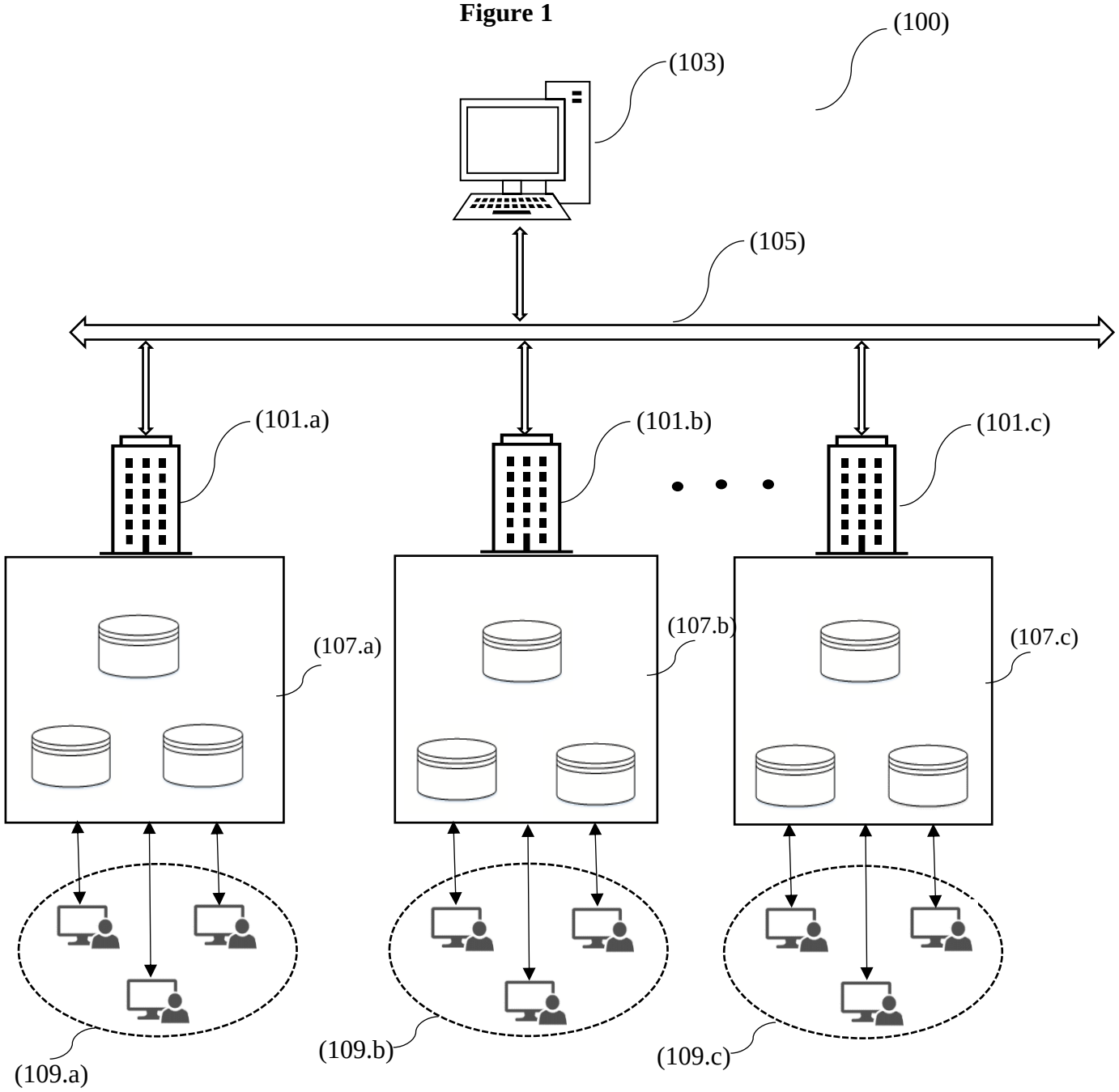


Figure 2

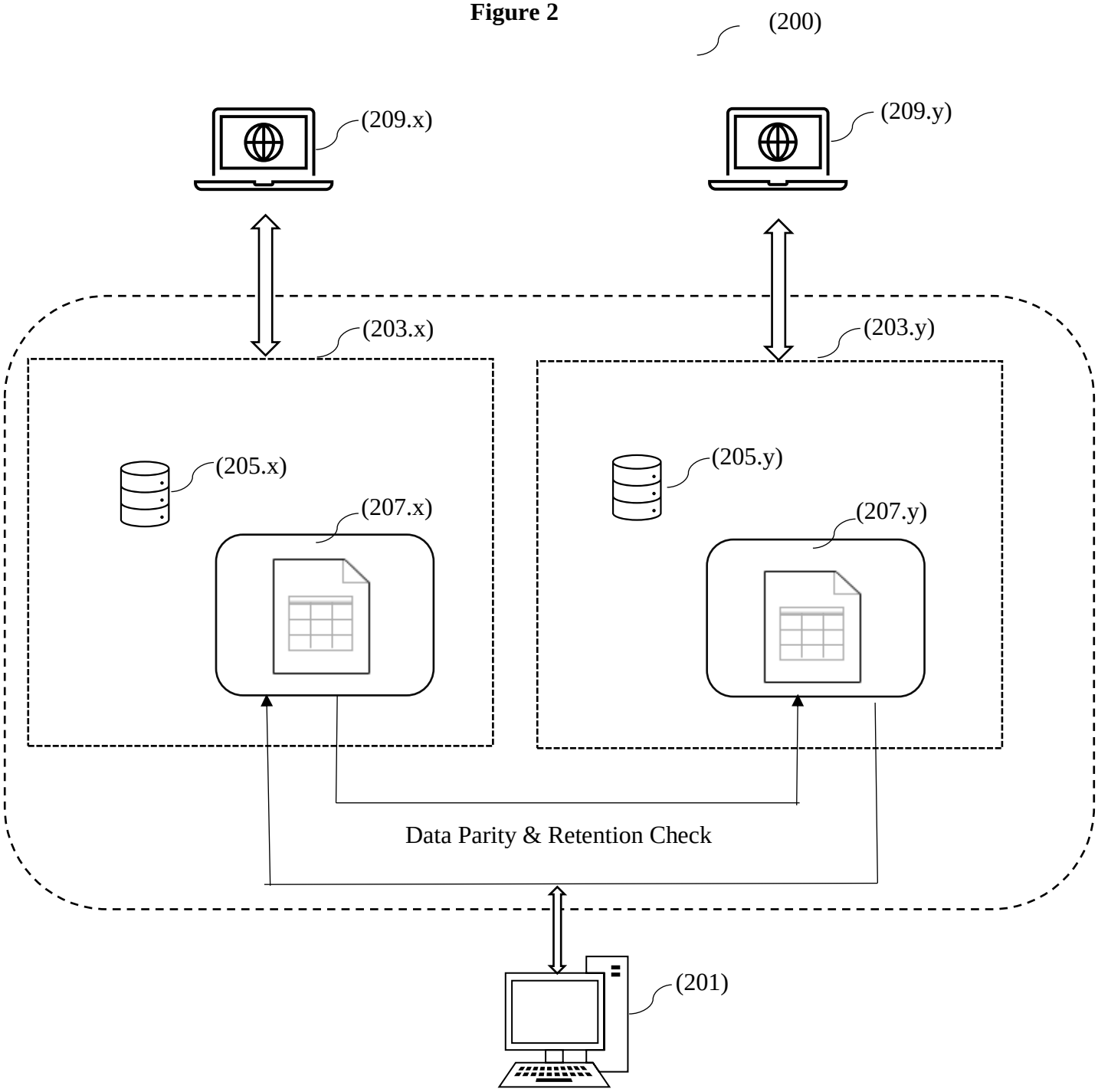

(200)
(209.x)
(209.y)
(203.x)
(203.y)
(205.x)
(205.y)
(207.x)
(207.y)
Data Parity & Retention Check
(201)

Figure 3

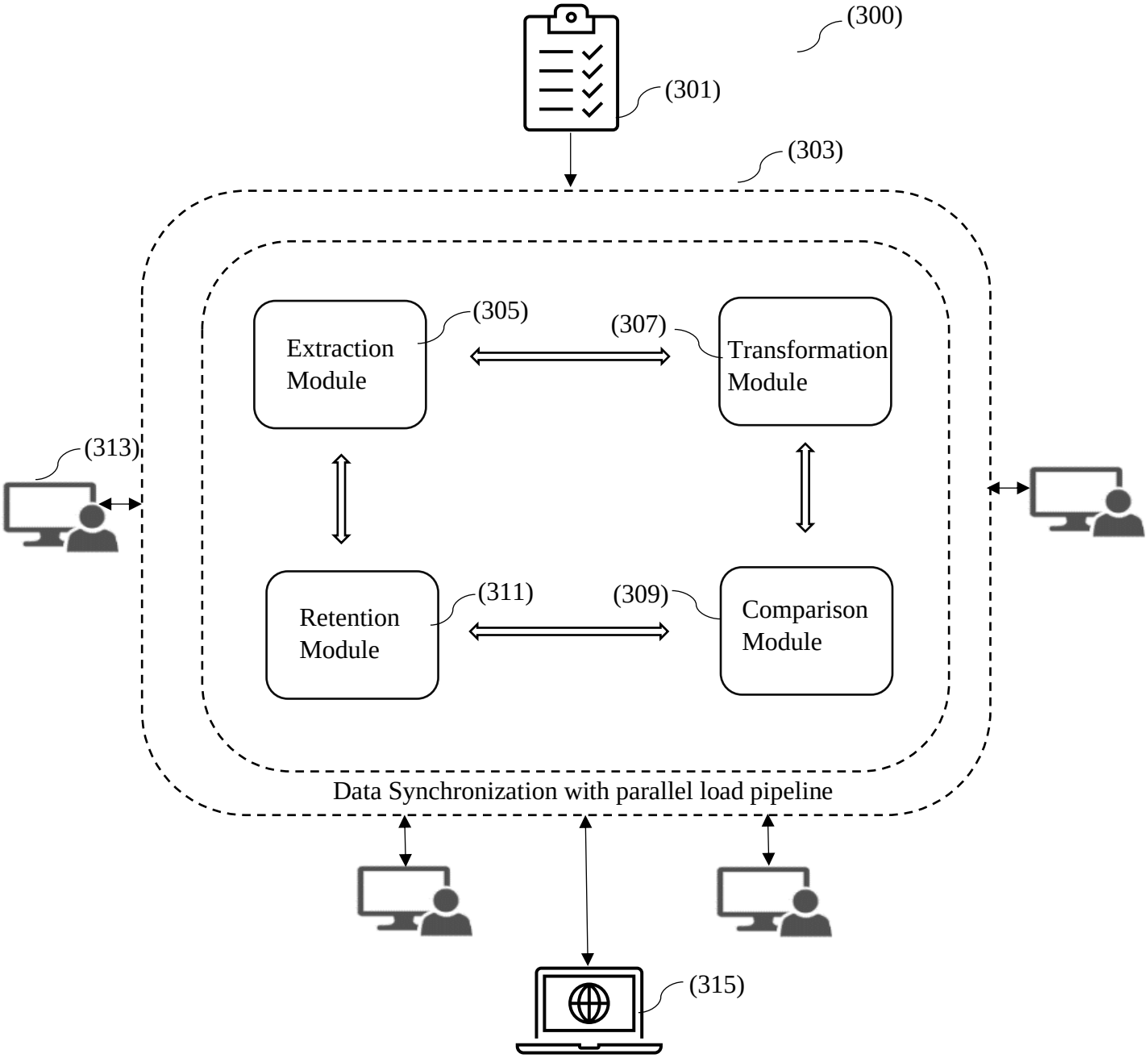

(300)
(301)
(303)
Extraction Module
(305)
(307)
Transformation Module
(313)
Retention Module
(311)
(309)
Comparison Module
Data Synchronization with parallel load pipeline
(315)

Figure 4

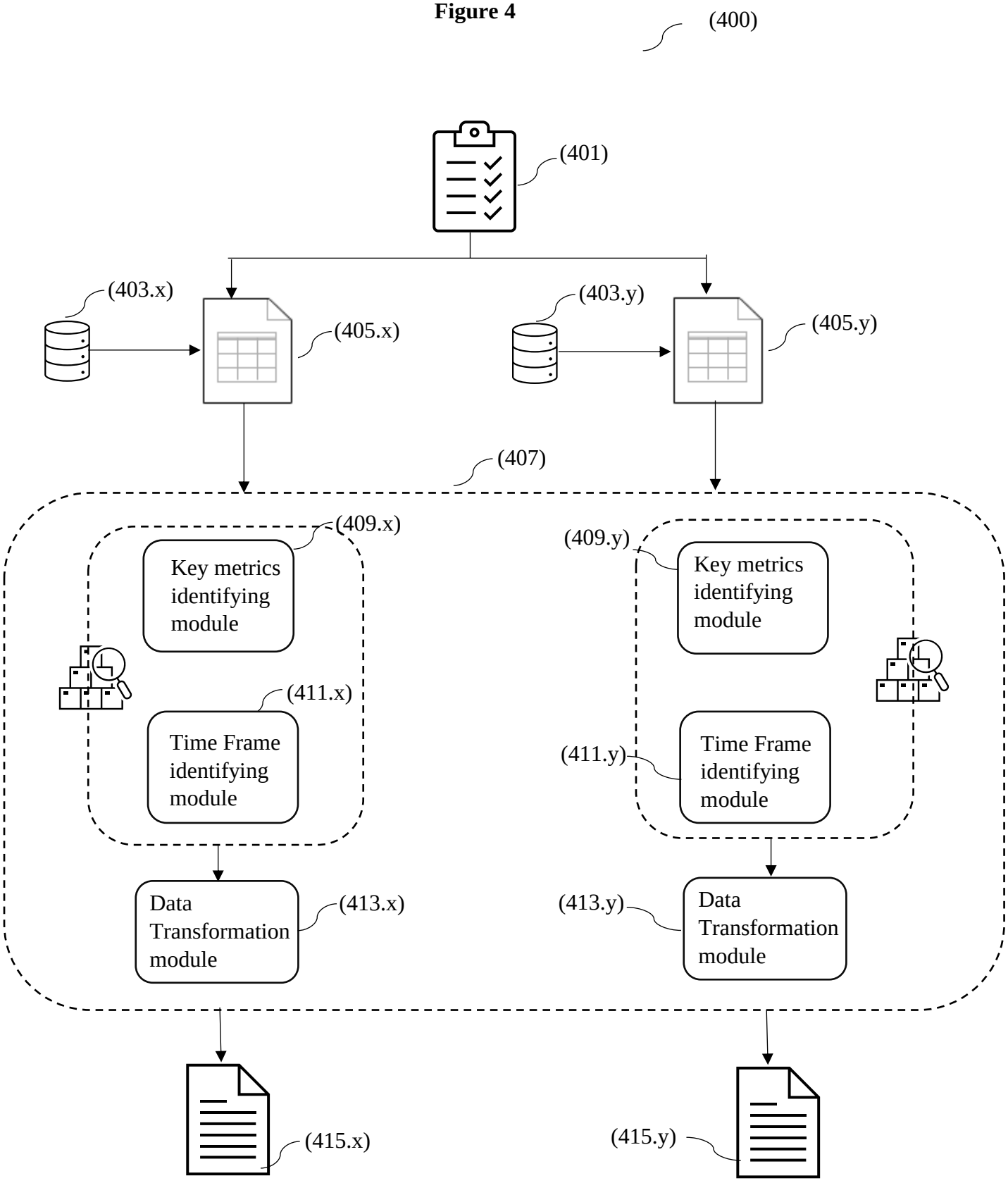

(400)
(401)
(403.x)
(405.x)
(403.y)
(405.y)
(407)
(409.x)
Key metrics identifying module
(411.x)
Time Frame identifying module
Data Transformation module
(413.x)
(409.y)
Key metrics identifying module
(411.y)
Time Frame identifying module
(413.y)
Data Transformation module
(415.x)
(415.y)

Figure 5

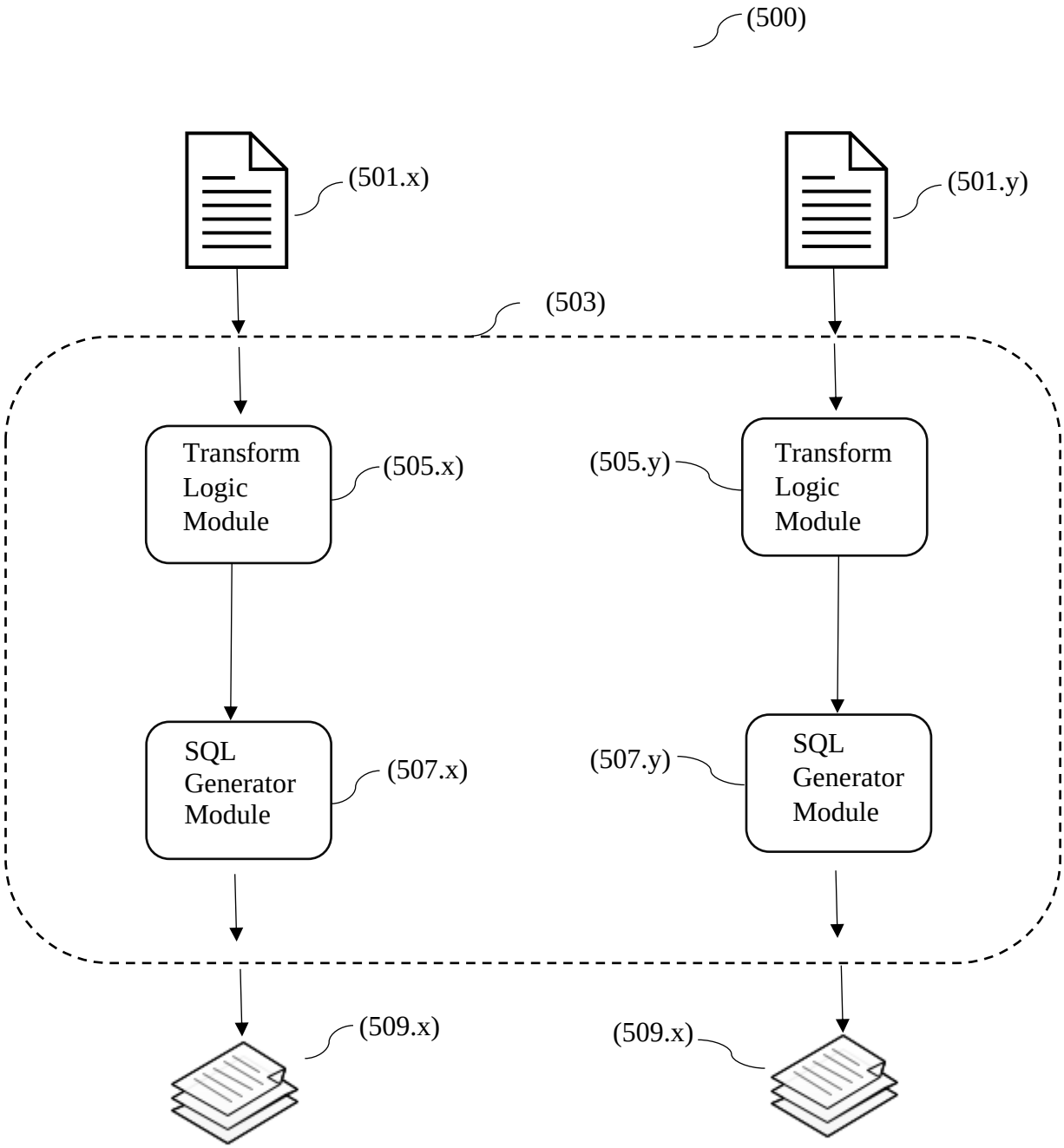

(500)
(501.x)
(501.y)
(503)
Transform Logic Module
(505.x)
(505.y)
Transform Logic Module
SQL Generator Module
(507.x)
(507.y)
SQL Generator Module
(509.x)
(509.x)

Figure 6

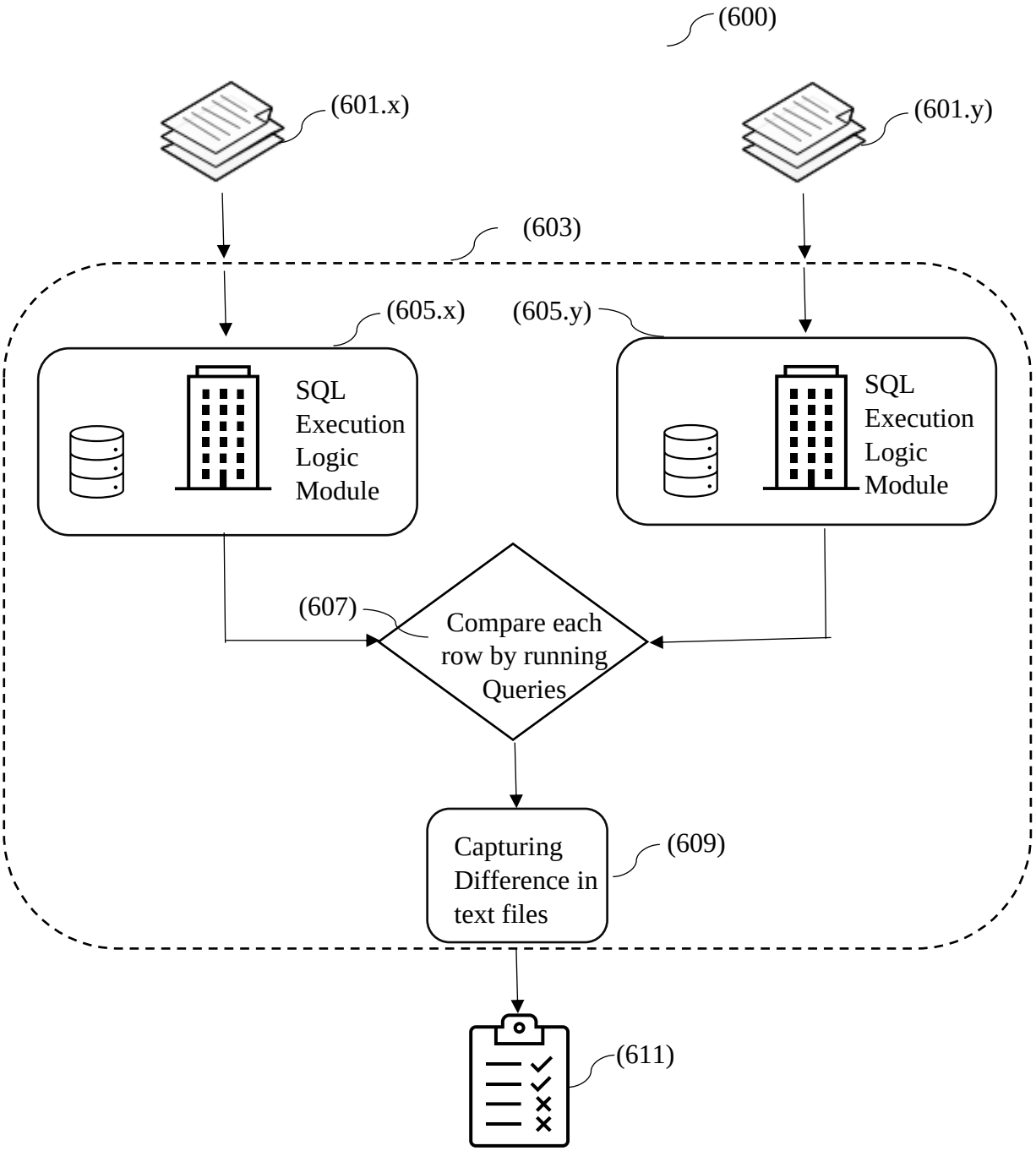

(600)
(601.x)
(601.y)
(603)
(605.x)
(605.y)
SQL Execution Logic Module
SQL Execution Logic Module
(607)
Compare each row by running Queries
Capturing Difference in text files
(609)
(611)

Figure 7

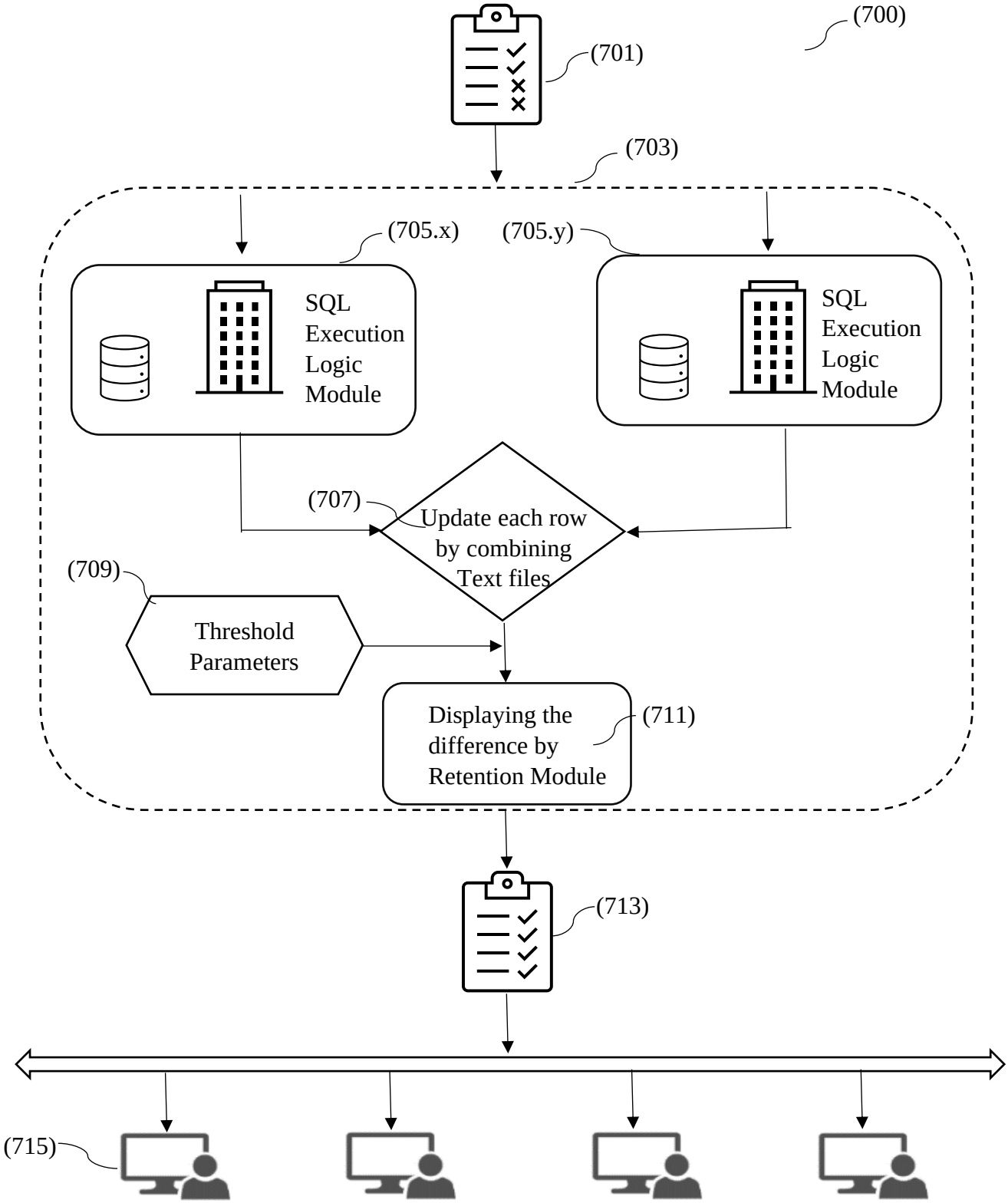

(700)
(701)
(703)
(705.x)
(705.y)
SQL Execution Logic Module
SQL Execution Logic Module
(707)
Update each row by combining Text files
(709)
Threshold Parameters
Displaying the difference by Retention Module
(711)
(713)
(715)

Figure 8

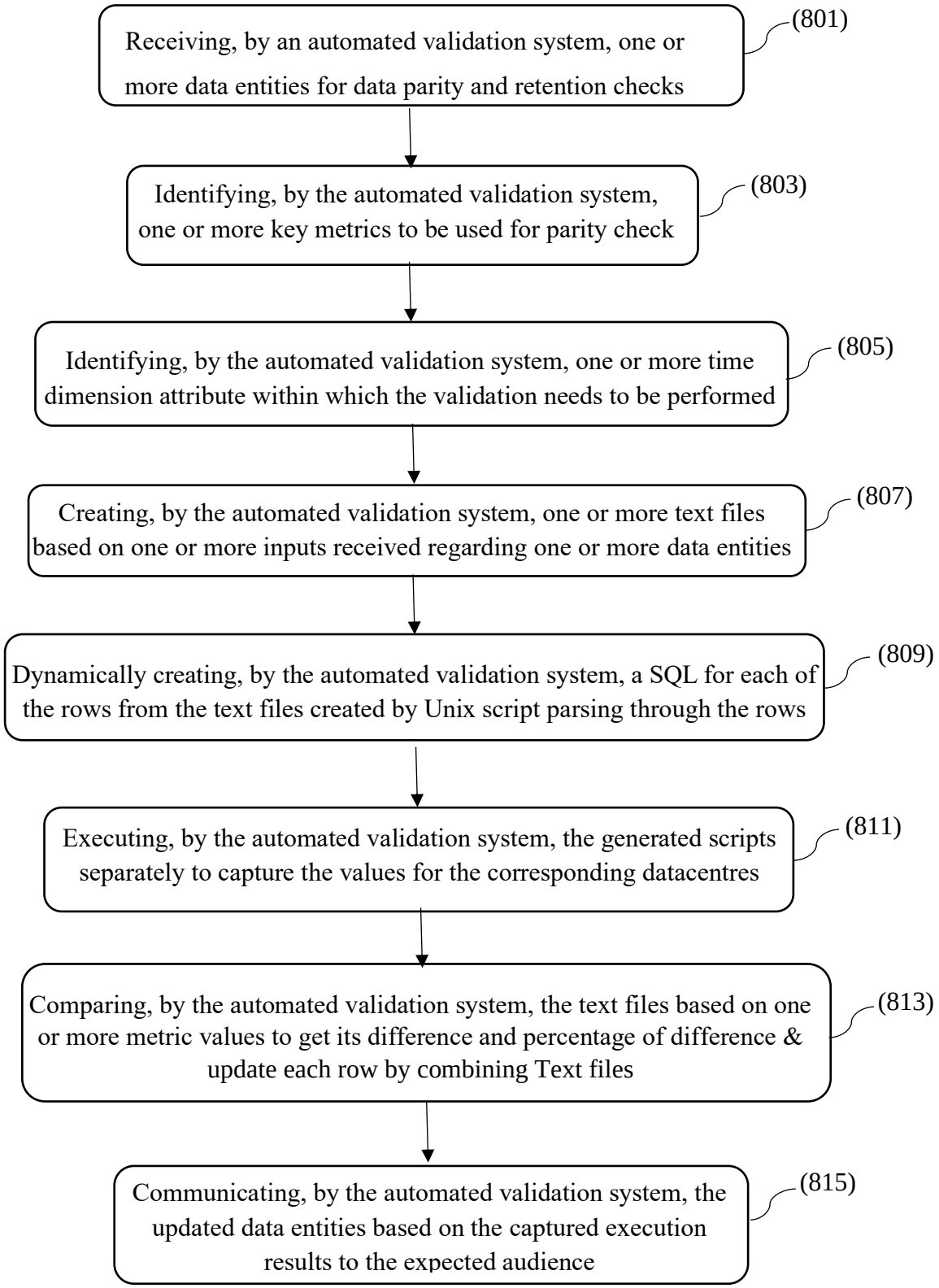
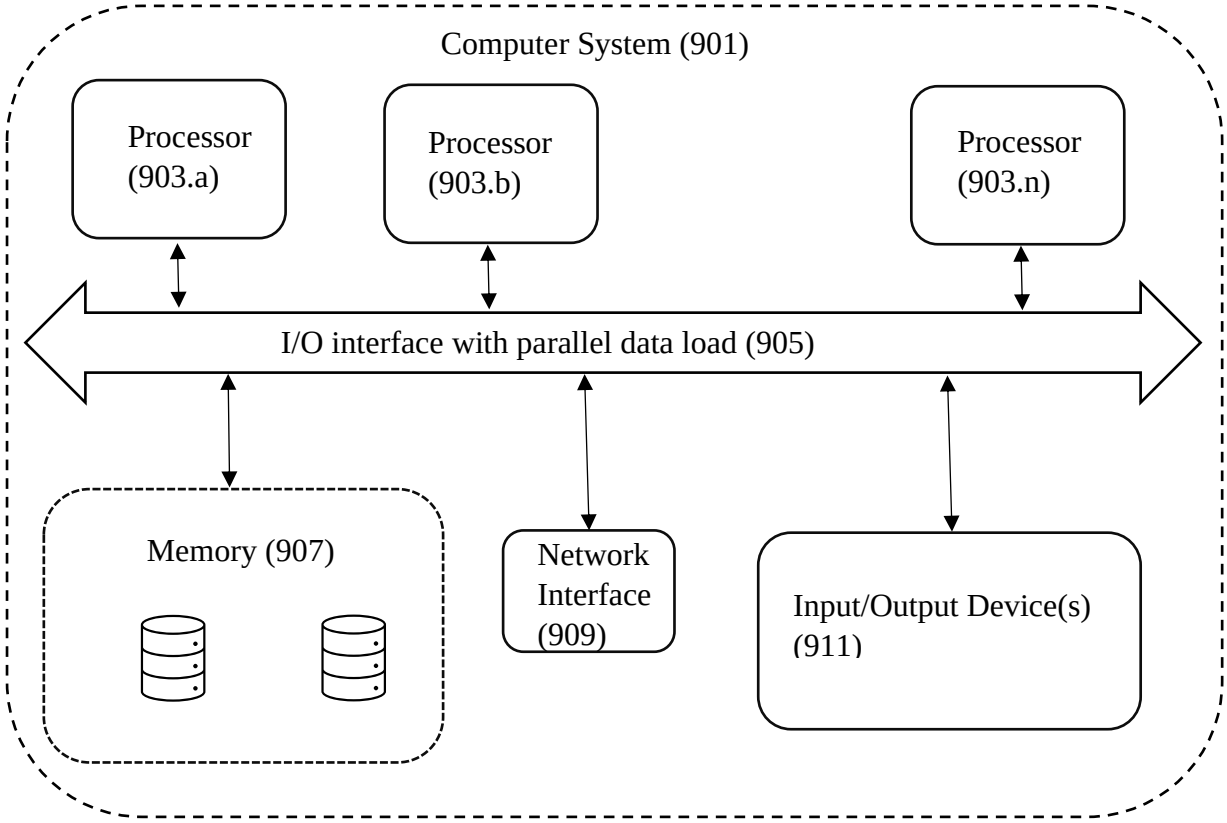

Receiving, by an automated validation system, one or more data entities for data parity and retention checks
(801)
Identifying, by the automated validation system, one or more key metrics to be used for parity check
(803)
Identifying, by the automated validation system, one or more time dimension attribute within which the validation needs to be performed
(805)
Creating, by the automated validation system, one or more text files based on one or more inputs received regarding one or more data entities
(807)
Dynamically creating, by the automated validation system, a SQL for each of the rows from the text files created by Unix script parsing through the rows
(809)
Executing, by the automated validation system, the generated scripts separately to capture the values for the corresponding datacentres
(811)
Comparing, by the automated validation system, the text files based on one or more metric values to get its difference and percentage of difference & update each row by combining Text files
(813)
Communicating, by the automated validation system, the updated data entities based on the captured execution results to the expected audience
(815)

Figure 9


Computer System (901)
Processor (903.a)
Processor (903.b)
Processor (903.n)
I/O interface with parallel data load (905)
Memory (907)
Network Interface (909)
Input/Output Device(s) (911)