

(51) International Patent Classification:
G06F 17/30 (2006.01)(21) International Application Number:
PCT/US2016/018237(22) International Filing Date:
17 February 2016 (17.02.2016)

(25) Filing Language: English

(26) Publication Language: English

(30) Priority Data:
15155691.7 19 February 2015 (19.02.2015) EP(71) Applicant: **MASTERCARD INTERNATIONAL INCORPORATED** [US/US]; 2000 Purchase Street, Purchase, New York 10577 (US).(72) Inventor: **GRENDON, Lauren**; 2000 Purchase Street, Purchase, New York 10577 (US).(74) Agent: **PANKA, Brian G.**; Harness, Dickey & Pierce, PLC, 7700 Bonhomme, Suite 400, St. Louis, Missouri 63105 (US).

(81) Designated States (unless otherwise indicated, for every kind of national protection available): AE, AG, AL, AM, AO, AT, AU, AZ, BA, BB, BG, BH, BN, BR, BW, BY, BZ, CA, CH, CL, CN, CO, CR, CU, CZ, DE, DK, DM, DO, DZ, EC, EE, EG, ES, FI, GB, GD, GE, GH, GM, GT, HN, HR, HU, ID, IL, IN, IR, IS, JP, KE, KG, KN, KP, KR, KZ, LA, LC, LK, LR, LS, LU, LY, MA, MD, ME, MG, MK, MN, MW, MX, MY, MZ, NA, NG, NI, NO, NZ, OM, PA, PE, PG, PH, PL, PT, QA, RO, RS, RU, RW, SA, SC, SD, SE, SG, SK, SL, SM, ST, SV, SY, TH, TJ, TM, TN, TR, TT, TZ, UA, UG, US, UZ, VC, VN, ZA, ZM, ZW.

(84) Designated States (unless otherwise indicated, for every kind of regional protection available): ARIPO (BW, GH, GM, KE, LR, LS, MW, MZ, NA, RW, SD, SL, ST, SZ, TZ, UG, ZM, ZW), Eurasian (AM, AZ, BY, KG, KZ, RU, TJ, TM), European (AL, AT, BE, BG, CH, CY, CZ, DE, DK, EE, ES, FI, FR, GB, GR, HR, HU, IE, IS, IT, LT, LU, LV, MC, MK, MT, NL, NO, PL, PT, RO, RS, SE, SI, SK, SM, TR), OAPI (BF, BJ, CF, CG, CI, CM, GA, GN, GQ, GW, KM, ML, MR, NE, SN, TD, TG).

Published:

— with international search report (Art. 21(3))

(54) Title: METHODS AND SYSTEMS FOR PROCESSING FILES

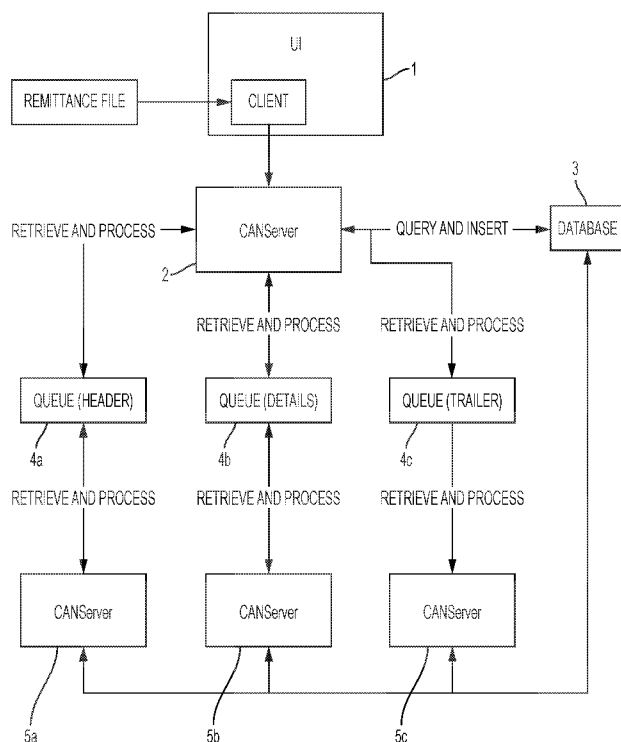


FIG. 1

(57) Abstract: The present disclosure relates to methods and systems for processing files containing record data. The method comprising splitting the record data into several record data parts and storing the record data parts in a database, for each of the record data parts, adding an entry identifying the record data part to one of one or more queues, for each of the entries added to the one or more queues, retrieving the entry and processing the record data part identified by the entry by one of a plurality of processing servers. The disclosure further describes comprising a database for a system for the same.

METHODS AND SYSTEMS FOR PROCESSING FILES

CROSS-REFERENCE TO RELATED APPLICATION

[0001] This application is a PCT International Application of, and claims priority to, European Patent Application No. 15155691.7 filed February 19, 2015. The entire disclosure of the above application is incorporated herein by reference.

FIELD

[0002] The present disclosure generally relates to information processing methods and systems. In particular, the present disclosure describes methods and systems for efficiently processing files containing record data.

BACKGROUND

[0003] This section provides background information related to the present disclosure which is not necessarily prior art.

[0004] File processing systems require the use of file processing servers which are capable of processing the file provided. In existing real time file processing systems, there is a requirement for providing a processing server which is capable of processing all the files. For example, when a large number of users upload files online for processing, there is considerable demand on the server which is processing the file. A file may comprise multiple records which are processed one after another by the same processor. Therefore, it is necessary to design and configure the system in advance taking into consideration the load factor that the processing server may encounter. However, it is not always possible to determine in advance the load on the processing server. Therefore, there is a requirement to provide scalability of servers to enable distributed processing while also maintaining integrity by avoiding multiple processing.

SUMMARY

[0005] This section provides a general summary of the disclosure, and is not a comprehensive disclosure of its full scope or all of its features. Aspects and embodiments of the disclosure are also set out in the accompanying claims.

[0006] The above objective is achieved by providing more processing servers where required and removing the processing servers when not required. In particular, the present disclosure provides a solution by providing a system and method where a raw file comprising multiple records is processed concurrently in a distributed architecture by multiple processing servers.

[0007] A first embodiment of the disclosure is related to a method of processing a file containing record data, comprising the steps of splitting the record data into several record data parts and storing the record data parts in a database. For each of the record data parts, an entry identifying the record data part is added to one of one or more queues. For each of the entries added to the one or more queues, the entry is retrieved and the record data part identified by the entry is processed by one of a plurality of processing servers.

[0008] By splitting the file and storing the parts individually in the database, it is possible to process the parts separately, thereby allowing multiple file processing at the same time. Adding an entry into a queue provides any processor the possibility to retrieve the data and process only the retrieved data from the database. The method according to the disclosure leads to a more scalable and distributed model in which servers can be added and removed depending on the load.

[0009] A database is an organized collection of data. Database management systems (DBMSs) are computer software applications that interact with the user, other applications, and the database itself to capture and analyze data. Existing general-purpose DBMSs are designed to allow the definition, creation, querying, update, and administration of databases. Administration of a database includes enforcing data security, maintaining data integrity, dealing with concurrency control, and recovering information that has been corrupted by some event such as an unexpected system failure. In the present disclosure, access by the processing servers to the data stored in the database is (typically) done through a DBMS. Storing the different data parts of the file to be processed in a database allows taking advantage of the above-mentioned features of existing DBMSs, to ensure integrity of data such that no two processors can access and process the same data from the database.

[0010] According to a second embodiment of the disclosure, in the method according to the first embodiment, an entry identifying a record data part is a primary key for the record data part stored in the database. The primary key by definition uniquely identifies entries of a

database and is therefore suitable for serving as an entry in the queue which identifies a record data part stored in the database. Furthermore, by using the primary key as an entry identifying the data, features of existing databases or database management systems are triggered by which transactional security is maintained by implicitly locking the unprocessed record in the database when retrieving to process thereby ensuring that no other server is processing at the same time.

[0011] According to a third embodiment of the disclosure, in the method according to any of the first or second embodiments, the file additionally contains header data comprising common data relating to several record data parts. The header data contains common information of the record data thereby reducing the amount of information on each record and thus the need to process it repeatedly.

[0012] According to a fourth embodiment of the disclosure, the method according to the third embodiment further comprises the steps of storing the header data in the database, adding one or more entries identifying the header data to one or more queues, and for each of the entries identifying the header data, retrieving the entry and processing the header data identified by the entry by one of the plurality of processing servers. By storing the header in the database, it is possible to process the header data and thereby allow header processing of multiple files at the same time. Adding an entry into a queue provides any processor the possibility to retrieve the data and process only the retrieved data from the database.

[0013] According to a fifth embodiment of the disclosure, in the method according to the fourth embodiment of the disclosure, an entry identifying header data is a primary key for that header data stored in the database. The primary key is suitable for serving as an entry in the queue which identifies header data stored in the database. Furthermore, by using the primary key as an entry identifying the data, features of existing databases or database management systems are triggered by which transactional security is maintained by implicitly locking the unprocessed header data in the database when retrieving to process thereby ensuring that no other server is processing at the same time.

[0014] According to a sixth embodiment of the disclosure, in the method according to the fourth or fifth embodiment, the one or more queues to which the one or more entries identifying the header data are added are separate from the one or more queues to which the entries identifying the record data parts are added. This ensures that a queue only has entries to particular data, thereby avoiding a processor from mistakenly processing other data from the

queue and only processing the data according to a predetermined order and maintaining the integrity of data.

[0015] According to a seventh embodiment of the disclosure, in the method according to any one of the third to sixth embodiment, the entries identifying the record data parts are added to the one or more queues only after the header data has been processed. This ensures that the data is processed in order so that all the common data is processed which is required for processing record data prior to processing the record data.

[0016] According to an eighth embodiment of the disclosure, the method according to the seventh embodiment, when embodying the fourth embodiment, further comprises the step of successful processing of the header data, transmitting the processed header data to the database with information indicating the start of the processing of the record data parts. This ensures that the system knows that the transmission of the entries identifying the record data parts to the queues can now start.

[0017] According to a ninth embodiment of the disclosure, in the method according to any one of the preceding embodiments, the file additionally contains trailer data comprising data for indicating whether the processing of the file is complete.

[0018] According to a tenth embodiment of the disclosure, the method according to the ninth embodiment of the disclosure further comprises the steps of storing the trailer data in the database, adding one or more entries identifying the trailer data to one or more queues, and for each of the entries identifying the trailer data, retrieving the entry and processing the trailer data identified by the entry by one of the plurality of processing servers. By storing the trailer in the database, it is possible to process the trailer data and thereby allow trailer processing of multiple files at the same time. Adding an entry into a queue provides any processor the possibility to retrieve the data and process only the retrieved data from the database.

[0019] According to an eleventh embodiment of the disclosure, in the method according to the tenth embodiment of the disclosure, an entry identifying trailer data is a primary key for that trailer data stored in the database. The primary key is suitable for serving as an entry in the queue which identifies trailer data stored in the database. Furthermore, by using the primary key as an entry identifying the data, features of existing databases or database management systems are triggered by which transactional security is maintained by implicitly

locking the unprocessed trailer data in the database when retrieving to process thereby ensuring that no other server is processing at the same time.

[0020] According to a twelfth embodiment of the disclosure, in the method according to the any one of the tenth or eleventh embodiments, the one or more queues to which the one or more entries identifying the trailer data are added are separate from the one or more queues to which the entries identifying the record data parts are added.

[0021] According to a thirteenth embodiment of the disclosure, in the method according to any one of the tenth to twelfth embodiments, when embodying the fourth embodiment, the one or more queues to which the one or more entries identifying the trailer data are added are separate from the one or more queues to which the one or more entries identifying the header data are added. This ensures that a queue only has entries to particular data, thereby avoiding a processor from mistakenly processing other data from the queue and only processing the data according to a predetermined order and maintaining the integrity of data.

[0022] According to a fourteenth embodiment of the disclosure, in the method according to any of the tenth to thirteenth embodiments of the disclosure, when embodying the third embodiment, the one or more entries identifying the trailer data are added to the one or more queues only after the header data has been processed.

[0023] According to a fifteenth embodiment of the disclosure, the method according to any one of the tenth to fourteenth embodiments of the disclosure, the trailer data identified by the one or more entries is processed by the processing servers only after the record data parts have been processed. Since the trailer data indicates the status of the processing, processing the trailer data after processing of the header data and record data ensures that the trailer data is processed in order to correctly indicate the processing status of the file.

[0024] According to a sixteenth embodiment of the disclosure, in the method according to the fifteenth embodiment, the trailer data identified by the one or more entries is processed by the processing servers after a predetermined time after adding the one or more entries identifying the trailer data to the one or more queues. This delay is provided so that all record data parts can be processed before the trailer data is processed in order to ensure that the status of processing is correctly provided.

[0025] According to a seventeenth embodiment of the disclosure, in the method according to any of the preceding embodiments, an entry from a queue is retrieved by a

processing server either by periodically polling the queue by the processing server or by the entry being pushed to the processing server. By providing a periodic polling of the queue, the processing servers can ensure that the data is processed immediately without a delay and also using the processing time efficiently. On the other hand, using the push mechanism by the queue to push the data to the processing server ensures that the processing server does not have to keep polling for a long duration.

[0026] According to an eighteenth embodiment of the disclosure, in the method according to any one of the preceding embodiments of the disclosure, the processing server retrieves an entry from the queue depending on whether or not the processor is currently free to process. By doing so, the processing servers can ensure that the data is processed utilizing the processing time efficiently.

[0027] According to a nineteenth embodiment of the disclosure, in the method according to any of the preceding embodiments, when a processing server retrieves an entry and while it processes the data identified by the entry, the database locks access to the data to avoid duplicate processing of the data. This provides for maintaining integrity of data such that no two processors can access and process the same data from the database at the same time.

[0028] According to a twentieth embodiment of the disclosure, in the method according to any of the preceding embodiments of the disclosure, the processing servers access the data stored in the database through a database management system (DBMS).

[0029] According to a twenty first embodiment of the disclosure, in the method according to the twentieth embodiment of the disclosure, the database management system is designed to allow enforcement of data security, maintaining data integrity, dealing with concurrency control, and/or recovering of information that has been corrupted by some event such as an unexpected system failure. Since many existing DBMSs already have such administration features, a system according to the disclosure which allows for data integrity and which ensures that no two processors can access and process the same data from the database at the same time can be implemented easily and in a cost-efficient manner, taking advantage of the features of such existing DBMSs. In particular, transactional security can be maintained by implicitly locking an unprocessed record in the database when retrieving it to process, thereby ensuring that no other server is processing it at the same time. As mentioned above, this can be

achieved, *e.g.*, by using an identifier added to the queues the primary key identifying the row and pointing to the record to be processed.

[0030] According to a twenty second embodiment of the disclosure, in the method according to any of the preceding embodiments of the disclosure, the database contains, for a data part in the database identified by an entry in a queue, information regarding the time when the most recent attempt to process the data part was made. This provides for further integrity of the data. The database maintains a retry count which counts the number of attempts and a pickup data value which maintains a timestamp with the date and time when the last attempt to access the data was made.

[0031] A twenty third embodiment of the disclosure is related to a computer program having instructions which when executed by a computing system cause the computing system to perform the method according to any of the first to twentieth embodiments.

[0032] A twenty fourth embodiment of the disclosure relates to a system for processing files comprising a database and a plurality of processing servers, the system being configured to perform the method according to any of the first to twentieth embodiments.

[0033] Further areas of applicability will become apparent from the description provided herein. The description and specific examples and embodiments in this summary are intended for purposes of illustration only and are not intended to limit the scope of the present disclosure. In addition, the above and other features will be better understood with reference to the followings Figures which are provided to assist in an understanding of the present teaching.

DRAWINGS

[0034] The drawings described herein are for illustrative purposes only of selected embodiments and not all possible implementations, and are not intended to limit the scope of the present disclosure.

[0035] With that said, the present disclosure is further described in the detailed description which follows, in reference to the noted plurality of drawings, by way of non-limiting examples of embodiments of the present disclosure, in which like reference signs represent like elements throughout the views of the drawings. In the following, the numbering of the embodiments does not coincide with the number of the embodiments in the above summary of the disclosure.

[0036] Fig. 1 illustrates an overall system implementing the present disclosure.

[0037] Fig. 2 illustrates a schematic flow diagram of an embodiment of the method of the present disclosure.

[0038] Corresponding reference numerals indicate corresponding parts throughout the several views of the drawings.

DETAILED DESCRIPTION

[0039] Exemplary embodiments will now be described more fully with reference to the accompanying drawings. The description and specific examples included herein are intended for purposes of illustration only and are not intended to limit the scope of the present disclosure.

[0040] The present disclosure will now be described in accordance with the system shown in Fig. 1. The system for file processing is provided with a user interface on a client computer (1). The user interface allows the user to upload a file to the system. The client devices may be connected to a server through the Internet or locally, and the file uploading may be implemented through an API on the client device or by another mode. However, a skilled person will understand that the file can be provided for processing through other means.

[0041] In the present embodiment, the file is a remittance file comprising multiple remittance records for performing transactions. A remittance file is merely used as an example as it illustrates a file with multiple records which require a high level of data integrity. However, a skilled person will understand that this system is not limited to such a file but any file having multiple records or which can be sub-divided into multiple parts so that each part can be processed independently. The file contains three components, namely header data, record data and trailer data, functions of which will be explained in detail subsequently. In the present example, the remittance file can contain up to twenty thousand records containing payment information.

[0042] The file is processed by a server (2) to split and store the components in a database (3). In the present embodiment, the server is a CANserver (Current Account Number Server); however any other server which performs the functions of the server can be used for the present disclosure. The database (3) is connected to the server (2) for storing the different components of the file and the multiple records containing payment information, for example.

[0043] One or more queues (4a, 4b, 4c) are provided and are connected to the server. A plurality of processing servers (5a, 5b, 5c) is connected to the queues and the database. A processing server could be the server (2) which also performs the function of a processing server. In the present disclosure, the processing servers implement the Java Messaging Service functionality which enables the processing servers to listen to the queues, *e.g.* by periodically polling the queues. However, the queue can also be configured to send the information periodically to the processing server by pushing.

[0044] Now the method implemented by the system will be explained in detail with reference to the flow diagram of Fig. 2.

[0045] A file is provided to the server through the user interface provided on the client system. The server then splits the file into raw data parts comprising one or more header data parts, several record data parts and one or more trailer data parts. The record data comprises multiple records. The record data parts along with the header data (parts) and trailer data (parts) are stored in the database (3).

Header data processing

[0046] The header data comprises common data relating to several record data parts. For example, in the remittance file, the header data may include common data such as currency details, date, etc. The server adds one or more entries identifying the header data to a header queue. In the present embodiment, the entry identifying header data is a primary key for that header data stored in the database and the primary key acts as a link to the particular data on the database. The header queue is one of the different queues, and is provided as a separate queue in the system. However, it is not necessary that a separate queue is provided but a single queue which performs the function of different queues can be provided instead.

[0047] The processing servers (5a, 5b, 5c) which process the header data listen to the header queue to see if there are new entries identifying header data. As soon as one of the processing servers retrieves an entry identifying header data, the header data which is identified by the entry is processed by the processing server. The processing server queries the database (3) for the appropriate header data relating to the entry and then processes the header data on the database (3). The database (3) identifies the header relating to the entry and returns the header data for processing. When the processing server processes the data identified by the entry, the

database (3) locks access to the data to avoid duplicate processing of the data. After completion of the processing of the header data, the server (2) writes back the processed header data to the database (3) and confirms the processing of the header data.

Record data processing

[0048] Following the successful processing of the header data, the server (2) retrieves from the database (3) the entries identifying the record data parts to be added to one or more queues. An entry identifying a record data part is a primary key for the record data part stored in the database (3). A record data part of the file in the particular embodiment is information concerning a transaction which needs to be processed. It may contain information such as the beneficiary's name, address, bank account and the amount to be remitted.

[0049] The server (2) then adds one or more entries identifying the record data part to a record queues (4b), which can also be called "detail queue". It must be noted that the entries identifying the record data parts are added to the queue and processed only after the header data has been processed.

[0050] In the present disclosure, an entry identifying a record data part is a primary key for that record data part stored in the database (3), and the primary key acts as a link to the particular data on the database (3). Similar to the header queue, the record queue is one of the different queues provided as a separate queue in the system. In the present embodiment, the record queue is separate from the header queue. However, it is not necessary that a separate queue is provided and a single queue which performs a function of different queues can be provided instead.

[0051] The processing servers (5b) which processes the record data parts listen to the record queue for entries identifying record data. One of the processing servers retrieves an entry from the queue depending on whether or not the processor is currently free to process. As soon as the processing server retrieves the entry identifying a record data part, the record data part which is identified by the entry is processed by the processing server. The processing server queries the database (3) for the appropriate record data part relating to the entry and then processes the record data part on the database (3). The database (3) identifies the record data part relating to the entry and returns the record data part for processing. More than one processing server can simultaneously read the entries from the queue, access the database (3) and

simultaneously process record data parts. This providing of multiple processing servers which can simultaneously read the database and process the data allows for scalability of processing by adding more processing servers when necessary.

[0052] The present disclosure also provides for maintaining integrity of data such that no two processors can access and process the same data from the database (3). When a processing server processes the data identified by the entry, the database (3) locks access to the data to avoid duplicate processing of the data. For further integrity of the data, the database (3) maintains a retry count which counts the number of attempts and a pickup data value which maintains a timestamp with the date and time when the last attempt to access the data was made.

Trailer data processing

[0053] Trailer data is provided which indicates whether the processing of the file is complete. After completion of the processing of the header data, the server (2) inserts processed header data to the database (3) to confirm the processing of the header data. Only after the header data has been processed, the server (2) adds one or more entries identifying the trailer data to a trailer queue (4c). In the present embodiment, the entry identifying trailer data is a primary key for that trailer data stored in the database (3), and the primary key acts as a link to the particular data on the database (3). The trailer queue is one of the different queues provided as a separate queue in the system. In the present embodiment, the trailer queue is separate from the header queue and the record queue. However, it is not necessary that a separate queue is provided, and a single queue which performs the function of different queues can be provided instead.

[0054] The processing servers (5a, 5b, 5c) which process the trailer data listen to the trailer queue for entries identifying trailer data. However, the processing servers retrieve and process the trailer data identified by the one or more entries only after a predetermined time after the entries are added to the queue. This delay is provided so that all record data parts can be processed before the trailer data is processed. One of the processing servers then retrieves an entry from the queue and processes trailer data. This means that on successful completion of the processing of all the record data parts, the trailer data is processed and the completed trailer data is then recorded on the database (3). In the absence of successful completion, the trailer data is marked as not completed. This will initiate further processing of the remaining uncompleted record processing by other processing servers.

[0055] In all the above, the processing servers retrieve the entry from the queue depending on whether or not the processor is currently free to process. The processing server is capable of processing all the data and does not have to be a standalone processing server for processing different data components such as header, record and trailer data. The same mechanism used for maintaining integrity of the data when processing the record data is also implemented for maintaining the integrity of other data types.

[0056] As a skilled person will understand, this method and system can be used on any file processing intensive task where the file can be sub-divided into multiple parts capable of being processed concurrently. The components of header, trailer and record are merely used as examples and other data types may be provided and implemented in the present disclosure.

[0057] The functions and/or steps and/or operations included herein, in some embodiments, may be described in computer executable instructions stored on a computer readable media (*e.g.*, in a physical, tangible memory, etc.), and executable by one or more processors. The computer readable media is a non-transitory computer readable storage medium. By way of example, and not limitation, such computer-readable media can include RAM, ROM, EEPROM, CD-ROM or other optical disk storage, magnetic disk storage or other magnetic storage devices, or any other medium that can be used to carry or store desired program code in the form of instructions or data structures and that can be accessed by a computer. Combinations of the above should also be included within the scope of computer-readable media.

[0058] Further, it should be appreciated that one or more aspects of the present disclosure transform a general-purpose computing device into a special-purpose computing device when configured to perform the functions, methods, and/or processes described herein.

[0059] With that said, exemplary embodiments are provided so that this disclosure will be thorough, and will fully convey the scope to those who are skilled in the art. Numerous specific details are set forth such as examples of specific components, devices, and methods, to provide a thorough understanding of embodiments of the present disclosure. It will be apparent to those skilled in the art that specific details need not be employed, that example embodiments may be embodied in many different forms and that neither should be construed to limit the scope of the disclosure. In some example embodiments, well-known processes, well-known device structures, and well-known technologies are not described in detail.

[0060] The present disclosure is not limited to the embodiment(s) described herein but can be amended or modified without departing from the scope of the present disclosure. Additionally, it will be appreciated that in embodiments of the present disclosure some of the above-described steps may be omitted and/or performed in an order other than that described. It will be appreciated that the payment card may be manufactured using steps such as, providing a card member; providing a plurality of programmable chips, at least one of which is releasably attached to the card member; each programmable chip has an associated personal identification number (PIN); and providing a control chip which is operable for selecting one of the programmable chips to be active during a transaction.

[0061] It will further be appreciated that elements of any embodiment disclosed herein may be combined interchangeably with elements of any other embodiment, except where such elements may be mutually exclusive. The above-described embodiments are, therefore, to be considered in all respects to be illustrative and not restrictive.

[0062] The terminology used herein is for the purpose of describing particular exemplary embodiments only and is not intended to be limiting. As used herein, the singular forms “a,” “an,” and “the” may be intended to include the plural forms as well, unless the context clearly indicates otherwise. The terms “comprises,” “comprising,” “including,” and “having,” are inclusive and therefore specify the presence of stated features, integers, steps, operations, elements, and/or components, but do not preclude the presence or addition of one or more other features, integers, steps, operations, elements, components, and/or groups thereof. As described above, the method steps, processes, and operations described herein are not to be construed as necessarily requiring their performance in the particular order discussed or illustrated, unless specifically identified as an order of performance. It is also to be understood that additional or alternative steps may be employed. As used herein, the term “and/or” includes any and all combinations of one or more of the associated listed items.

[0063] When a feature is referred to as being “on,” “engaged to,” “connected to,” “coupled to,” “attached to,” “associated with,” “included with,” or “in communication with” another feature, it may be directly on, engaged, connected, coupled, attached, associated, included, or in communication to or with the other feature, or intervening features may be present.

CLAIMS

What is claimed is:

1. A method of processing a file containing record data, the method comprising:
splitting the record data into several record data parts and storing the record data parts in a database;
for each of the record data parts, adding an entry identifying the record data part to one of one or more queues; and
for each of the entries added to the one or more queues, retrieving the entry and processing the record data part identified by the entry by one of a plurality of processing servers.
2. The method of claim 1, wherein an entry identifying a record data part is a primary key for the record data part stored in the database.
3. The method of claim 1, wherein the file additionally contains header data comprising common data relating to several record data parts; and
further comprising:
storing the header data in the database;
adding one or more entries identifying the header data to one or more queues; and
for each of the entries identifying the header data, retrieving the entry and processing the header data identified by the entry by one of the plurality of processing servers.
4. The method of claim 3, wherein an entry identifying header data is a primary key for that header data stored in the database.
5. The method of claim 3, wherein the entries identifying the record data parts are added to the one or more queues only after the header data has been processed.

6. The method of claim 5, further comprising, on successful processing of the header data, transmitting the processed header data to the database with information indicating the start of the processing of the record data parts.

7. The method of claim 3, wherein the file additionally contains trailer data comprising data for indicating whether the processing of the file is complete; and further comprising:

storing the trailer data in the database;

adding one or more entries identifying the trailer data to one or more queues; and

for each of the entries identifying the trailer data, retrieving the entry and processing the trailer data identified by the entry by one of the plurality of processing servers;

wherein the one or more entries identifying the trailer data are added to the one or more queues only after the header data has been processed.

8. The method of claim 3, wherein the one or more queues to which the one or more entries identifying the header data are added are separate from the one or more queues to which the entries identifying the record data parts are added.

9. The method of claim 1, wherein the file additionally contains trailer data comprising data for indicating whether the processing of the file is complete; and further comprising:

storing the trailer data in the database;

adding one or more entries identifying the trailer data to one or more queues; and

for each of the entries identifying the trailer data, retrieving the entry and processing the trailer data identified by the entry by one of the plurality of processing servers.

10. The method of claim 9, wherein an entry identifying trailer data is a primary key for that trailer data stored in the database.

11. The method of claim 9, wherein the trailer data identified by the one or more entries is processed by the processing servers only after the record data parts have been processed, and after a predetermined time after adding the one or more entries identifying the trailer data to the one or more queues.

12. The method of claim 9, wherein the one or more queues to which the one or more entries identifying the trailer data are added are separate from the one or more queues to which the entries identifying the record parts are added.

13. The method of claim 9, wherein the one or more queues to which the one or more entries identifying the trailer data are added are separate from the one or more queues to which the one or more entries identifying the header data are added.

14. The method of claim 1, wherein, when a processing server retrieves an entry and while it processes the data identified by the entry, the database locks access to the data to avoid duplicate processing of the data.

15. The method of claim 1, wherein the processing servers access the data stored in the database through a database management system which is designed to allow enforcement of data security, maintain data integrity, deal with concurrency control, and/or recover information that has been corrupted by some event such as an unexpected system failure.

16. The method of claim 1, wherein the database contains, for a data part in the database identified by an entry in a queue, information regarding the time when the most recent attempt to process the data part was made.

17. The method of claim 1, wherein an entry from a queue is retrieved by a processing server either by periodically polling the queue by the processing server or by the entry being pushed to the processing server.

18. The method of claim 1, wherein the processing server retrieves an entry from the queue depending on whether or not the processor is currently free to process.

19. A non-transitory computer readable storage media including executable instructions, which when executed by at least one processor, cause the at least one processor to:
split record data into several record data parts and store the record data parts in a database;

for each of the record data parts, add an entry identifying the record data part to one of one or more queues; and

for each of the entries added to the one or more queues, retrieve the entry and process the record data part identified by the entry by one of a plurality of processing servers.

20. A system for processing a file comprising a database and a plurality of processing servers, the system configured to:

split record data into several record data parts and store the record data parts in the database;

for each of the record data parts, add an entry identifying the record data part to one of one or more queues; and

for each of the entries added to the one or more queues, retrieve the entry and process the record data part identified by the entry by one of the plurality of processing servers.

REPLACEMENT SHEET

1/2

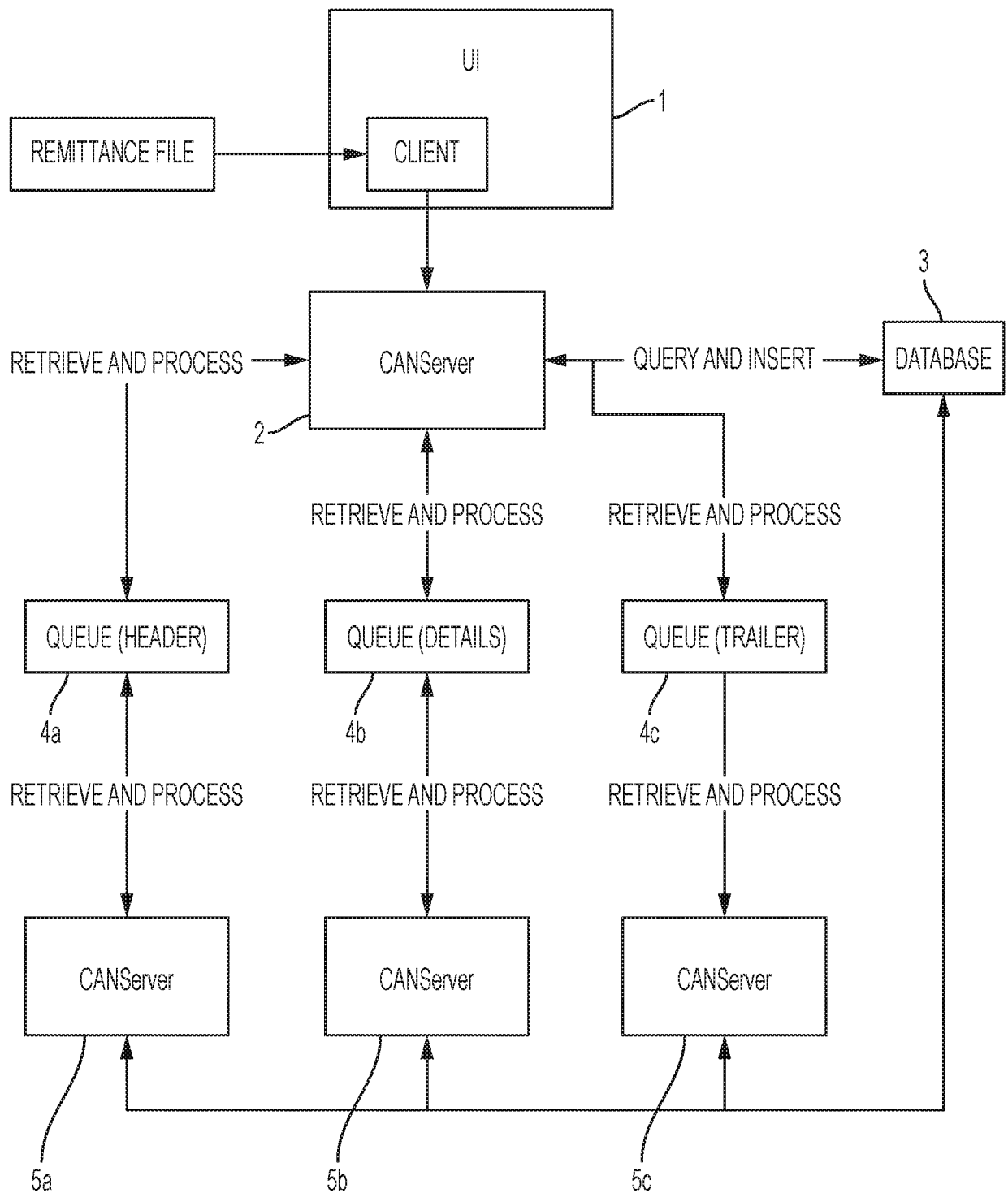


FIG. 1

REPLACEMENT SHEET

2/2

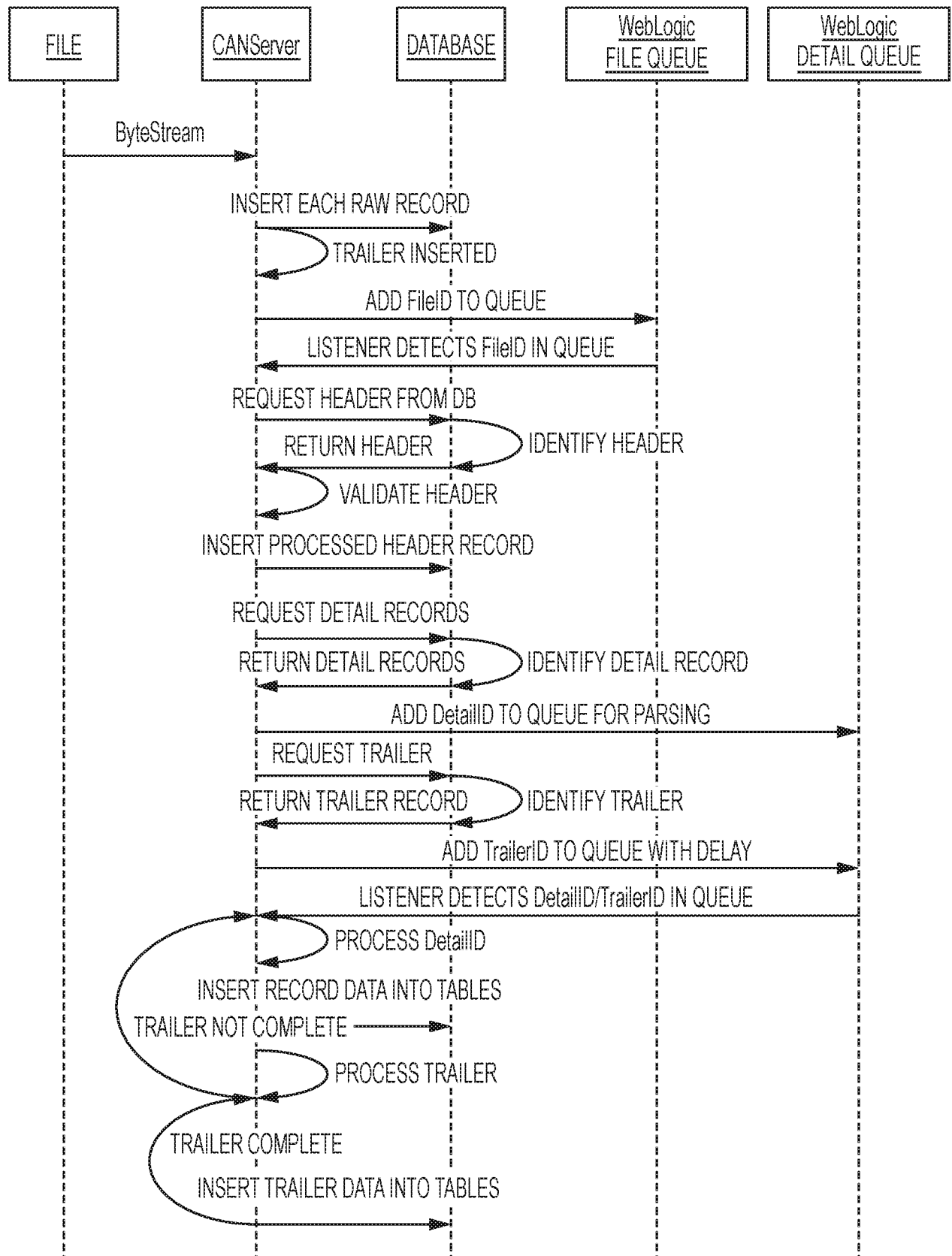


FIG. 2

A. CLASSIFICATION OF SUBJECT MATTER**G06F 17/30(2006.01)i**

According to International Patent Classification (IPC) or to both national classification and IPC

B. FIELDS SEARCHED

Minimum documentation searched (classification system followed by classification symbols)

G06F 17/30; H03M 7/30; G06F 15/16; G06F 17/40

Documentation searched other than minimum documentation to the extent that such documents are included in the fields searched

Korean utility models and applications for utility models

Japanese utility models and applications for utility models

Electronic data base consulted during the international search (name of data base and, where practicable, search terms used)

eKOMPASS(KIPO internal) & Keywords: splitting, entry, processing servers, primary key, queues, and similar terms.

C. DOCUMENTS CONSIDERED TO BE RELEVANT

Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
X	US 2012-0158650 A1 (MIHNEA ANDRE et al.) 21 June 2012 See paragraphs [0044], [0046], [0053], [0062]-[0064], [0066], [0072]-[0073], [0077]-[0079], and [0094]; claims 1, 15, and 22; and figures 1-5.	1-20
A	US 2013-0318034 A1 (JASKARAN SINGH BAWA et al.) 28 November 2013 See paragraphs [0024] and [0028]; and figures 2A and 3.	1-20
A	US 2009-0265305 A1 (ERIC L. BARSNESS et al.) 22 October 2009 See paragraphs [0053]-[0055] and figure 6.	1-20
A	US 2011-0279294 A1 (ARUN KUMAR SAGAR) 17 November 2011 See paragraphs [0031]-[0033] and figures 4-5.	1-20
A	WO 2014-165283 A1 (VULCAN TECHNOLOGIES LLC) 09 October 2014 See page 20, line 27 - page 22, line 22; and figure 9.	1-20



Further documents are listed in the continuation of Box C.



See patent family annex.

* Special categories of cited documents:

"A" document defining the general state of the art which is not considered to be of particular relevance

"E" earlier application or patent but published on or after the international filing date

"L" document which may throw doubts on priority claim(s) or which is cited to establish the publication date of another citation or other special reason (as specified)

"O" document referring to an oral disclosure, use, exhibition or other means

"P" document published prior to the international filing date but later than the priority date claimed

"T" later document published after the international filing date or priority date and not in conflict with the application but cited to understand the principle or theory underlying the invention

"X" document of particular relevance; the claimed invention cannot be considered novel or cannot be considered to involve an inventive step when the document is taken alone

"Y" document of particular relevance; the claimed invention cannot be considered to involve an inventive step when the document is combined with one or more other such documents, such combination being obvious to a person skilled in the art

"&" document member of the same patent family

Date of the actual completion of the international search

02 June 2016 (02.06.2016)

Date of mailing of the international search report

07 June 2016 (07.06.2016)

Name and mailing address of the ISA/KR

International Application Division

Korean Intellectual Property Office

189 Cheongsa-ro, Seo-gu, Daejeon, 35208, Republic of Korea

Facsimile No. +82-42-481-8578

Authorized officer

NHO, Ji Myong

Telephone No. +82-42-481-8528



INTERNATIONAL SEARCH REPORT

Information on patent family members

International application No.

PCT/US2016/018237

Patent document cited in search report	Publication date	Patent family member(s)	Publication date
US 2012-0158650 A1	21/06/2012	None	
US 2013-0318034 A1	28/11/2013	US 2013-318031 A1 US 8738649 B2 US 8738650 B2 WO 2013-175308 A1 WO 2013-175308 A9	28/11/2013 27/05/2014 27/05/2014 28/11/2013 09/01/2014
US 2009-0265305 A1	22/10/2009	US 8195642 B2	05/06/2012
US 2011-0279294 A1	17/11/2011	US 8427348 B2 WO 2011-143585 A1	23/04/2013 17/11/2011
WO 2014-165283 A1	09/10/2014	US 2014-337321 A1	13/11/2014