Method and System for Adaptively Processing a Request for Data

FIELD OF INVENTION

[1] The present disclosure relates broadly, but not exclusively, to methods and systems for adaptively processing a request for data.

BACKGROUND

[2] Risk management for a platform offering various services and/or products for sale typically involves maintaining data points related to users who are using the platform. These data points can then be used to establish rules for fraud prevention. Some examples of these data points may be the number of devices from which the user has successfully interacted with the platform, the number of users using a same device or account to interact with the platform, and other similar types of data. To detect fraud, it is desirable to assess the data over various time windows in real-time. These windows can be as small as a minute and can be as big as a user's lifetime in the platform's ecosystem. A risk management system can collect this data and aggregate it over several time windows using, for example, an aggregation engine.

[3] One known type of aggregation engine ensures the real-time calculation of aggregates by maintaining pre-computed data in fixed time intervals. For example, these kinds of systems can maintain data in time intervals of a minute, an hour, eight hours, twelve hours and so on. This approach is generally efficient but suffers from the drawback that it cannot furnish data in arbitrary time intervals like 8 hours 30 minutes and is limited to calculating aggregates in a single time zone only. Storing these aggregates for every time zone that a platform operates in would be costly and operationally challenging.

[4] The second kind of aggregation engine maintains the data in relational database tables with sharding, which is a method of splitting and storing a single logical dataset into, for example, multiple logical tables. This approach works fine if the interval in which data

is being requested lies in a couple of shards. If that is not the case, however, it would be very difficult to process a large number of data points in a reasonable time.

[5] A need therefore exists to provide methods and systems that seek to overcome or at least minimize the above mentioned challenges.

SUMMARY

[6] According to a first aspect of the present disclosure, there is provided a method for adaptively processing a request for data, comprising: comparing, by a processor, a first time range with a second time range, the first time range being based on a start time and an end time indicated in a request for data associated with the start time and the end time, and the second time range being associated with a first database or a second database; determining, by the processor, one or more sub time ranges for retrieving the data, the one or more sub time ranges being based on the first time range and the comparison; assessing, by the processor, whether to retrieve data associated with each of the one or more sub time ranges from the first database or the second database based on the comparison; and retrieving the data based on the assessment.

[7] According to a second aspect of the present disclosure, there is provided a system for identifying a match with a product, comprising: at least one processor; and at least one memory including computer program code; the at least one memory and the computer program code configured to, with the at least one processor, cause the system at least to: compare, by a processor, a first time range with a second time range, the first time range being based on a start time and an end time indicated in a request for data associated with the start time and the end time, and the second time range being associated with a first database or a second database; determine, by the processor, one or more sub time ranges for retrieving the data, the one or more sub time ranges being based on the first time range and the comparison; and assess, by the processor, whether to retrieve data associated with each of the one or more sub time ranges from the first database or the second database based on the comparison; and retrieving the data based on the assessment.

BRIEF DESCRIPTION OF THE DRAWINGS

[8] Embodiments and implementations are provided by way of example only, and will be better understood and readily apparent to one of ordinary skill in the art from the following written description, read in conjunction with the drawings, in which:

[9] Fig. 1 illustrates a system for adaptively processing a request for data according to various embodiments of the present disclosure.

[10] Fig. 2 is a schematic diagram of a data processing server, according to various embodiments of the present disclosure.

[11] Fig. 3 is an overview of a process for adaptively processing a request for data, according to various embodiments.

[12] Fig. 4A depicts an example illustration of how performance is improved by the present invention according to various embodiments.

[13] Fig. 4B depicts an example illustration of how aggregates may be stored according to various embodiments.

[14] Fig. 4C depicts an example illustration of an amount of memory required in kilobytes for a system configured to retrieve a data point within any time falling within a past one year according to various embodiments.

[15] Fig. 5 illustrates an example illustration of an expression tree framework for adaptively processing a request for data according to various embodiments.

[16] Figs. 6A – 6E illustrate an example illustration of a flowchart for adaptively processing a request for data according to various embodiments.

[17] Fig. 7 illustrates an example flow diagram of a method for adaptively processing a request for data according to various embodiments.

[18] Fig. 8A is a schematic block diagram of a general purpose computer system upon which the data processing server of Fig. 2 can be practiced.

[19] Fig. 8B is a schematic block diagram of a general purpose computer system upon which a combined transaction processing and data processing server of Fig. 1 can be practiced.

[20] Fig. 9 shows an example of a computing device to realize the transaction processing server shown in Fig. 1.

[21] Fig. 10 shows an example of a computing device to realize the data processing server shown in Fig. 1.

[22] Fig. 11 shows an example of a computing device to realize a combined transaction processing and data processing server shown in Fig. 1.

[23] Skilled artisans will appreciate that elements in the figures are illustrated for simplicity and clarity and have not necessarily been depicted to scale. For example, the dimensions of some of the elements in the illustrations, block diagrams or flowcharts may be exaggerated in respect to other elements to help to improve understanding of the present embodiments.

DETAILED DESCRIPTION

Terms Description

[24] A platform refers to a set of technologies that is used as a base for facilitating exchanges between two or more interdependent groups, for example between a user (of a product or service) and a provider (of the product or service). For example, a platform may offer a service offered by a provider such as a ride, delivery, online shopping, insurance, and other similar services to a requestor. The user can typically access the platform via a website, an application, or other similar methods.

[25] An aggregation engine or aggregation server refers to an engine or server for data and information retrieval on demand over relational and non-relational distributed data sources.

[26] For example, Kafka™ has an aggregation engine which has a concept of windows to collect aggregations. These windows are similar to time intervals storing pre-computed data. So, it falls under the first category of aggregation engine. There is a feature called continuous aggregates in TimescaleDB™, a time-series database, which also falls under the first category of aggregation engine, wherein raw data (e.g. data that is not yet processed into aggregated data for a time range) may be stored in the time-series database.

[27] A NoSQL database (aka “not only Structured Query Language”) are non-tabular databases and store data differently than relational tables. NoSQL databases come in a variety of types based on their data model. The main types are document, key-value, wide-column, and graph. They provide flexible schemas and scale easily with large amounts of data and high user loads. An example of a NoSQL database is Aerospike™ which is a flash memory and in-memory open source distributed key value NoSQL database management system. Aggregated data which is data that has been processed for a time range may be stored in the NOSQL database. Examples of such aggregated data may be the number of successful transactions for a user within a time range, the number of devices from which the user has done a successful transaction within a time range, the number of users using a same card to transact on the platform within a time range, and other similar types of data. The time range may vary depending on application, such as for example 30 days for daily aggregates, 12 months for monthly aggregates, or other similar time ranges.

[28] An expression tree refers to a representation of expressions arranged in a tree-like data structure. In other words, it is a tree with leaves as operands of the expression and nodes contain the operators. Similar to other data structures, data interaction is also possible in an expression tree. Expression trees are mainly used for analyzing, evaluating and modifying expressions, especially complex expressions. In the present disclosure, an expression tree may be implemented by an algorithm and utilized for adaptively processing a request for data. The expression tree may comprise a plurality of nodes. Each node may be connected to one or more other nodes by an edge or link. A node may be connected from a downstream position to another node in an upstream position. In this case, the node at the downstream position may be termed as a child node, while the node at the upstream position may be termed as a parent node. Each node may be configured

to perform an operation, such as for example collection of data from a database, addition of data (e.g. collected by two or more children nodes), subtraction of data (e.g. subtraction of data collected by a child node from data collected by another child node), and other similar operations.

[29] Embodiments of the present disclosure are able to provide ad hoc aggregations optimally by holding pre-calculated aggregates in a first database (for example, a NoSQL database such as Aerospike™) and raw events in a second database (for example, a time series database such as TimescaleDB™). A fragmentation logic is then run to break an original time range for which data is required into several sub time ranges such that as much of the data as possible can be obtained from pre-calculated aggregates from the NoSQL database and the minimum amount of data has to be fetched by doing operations on raw data. For example, if data is required for a 90 day period ending on 2021-08-25 00:00:00, assuming the current date is 2021-10-01 00:00:00, a fragmentation tree such as shown in illustration 300 of Figure 3 may be constructed. This tree breaks down the execution into 3 nodes. A first node 302 collects data from pre-computed monthly aggregates stored in, for example, a NoSQL database for the two complete months within the 90 day period (2021-06-01 to 2021-07-31). A second node 304 collects raw data for the initial four days (2021-05-27 to 2021-05-31) from, for example, a time-series database. The third node 306 subtracts time series raw data for the last six days in the last incomplete month (ending at the end time of the 90 day period), the time series raw data being collected by a node 308, from a pre-computed monthly aggregate stored in the NoSQL database for the August month, the pre-computed monthly aggregate being collected by a node 310. Rather than fetching data for the 25 days (2021-08-01 to 2021-08-25) directly from a time-series database (which is an expensive operation requiring iterations of processing over the dataset stored in the time-series database), the third node 306 does the subtraction operation instead which is advantageously less intensive in terms of processing power and amount of time required. Finally, all the data points are added via node 312 to arrive at the aggregated data for the 90 day period ending on 2021-08-25 00:00:00.

[30] An Aerospike™ database may be used as a first database for being able to provide Dynamic Random Access Memory (DRAM)-like performance with Solid State Drives, since storing such huge amounts of data in memory (for example in RAM) would be expensive. As shown in illustration 400 of Figure 4A, it is possible to achieve a sub-one-

millisecond latency for 97% of data requests and sub-eight-millisecond latency for 99.83% of data requests in Aerospike™.

[31] Illustration 402 of Figure 4B shows an example data structure for a data point stored for a user who has done one successful transaction of 100 Singapore Dollars on a platform. Daily aggregates (e.g. data that is aggregated on a daily basis) may be represented in a yy-mm-dd format as shown at 404, which indicates the date of 8 December 2021 as “211208”, while monthly aggregates (e.g. data that is aggregated on a monthly basis) may be represented in a yy-mm format as shown at 406, which indicates the same date of 8 December 2021 as “2112”. One advantage of storing data in this hierarchical fashion is that records can be reused to store and map large numbers of data points. For example, the records as shown in illustration 402 may be used to fetch information about how many transactions a user has attempted in a given period, as well as tell us how many successful transactions a user has done in Singapore Dollars.

[32] Monthly aggregates for the past 12 months (for example) and daily aggregates for the past 1 month (for example) may be maintained in an Aerospike database. These aggregates may then be automatically purged or deleted from the database after the duration (e.g. 12 months and 1 month respectively) has passed. Therefore, in some embodiments, an algorithm may be utilized for adaptively processing a request for data such that a Day Purge Duration is set as 30 days and a Month Purge Duration is set as 12 months. This algorithm works very well to retrieve a data point within any time range falling in the last 1 year. The Aerospike™ record size varies according to the number of entities and the time ranges stored in the nested map. An Aerospike™ database, as of today, has a record size limit of 10 megabits (MB). Therefore, this algorithm can be readily extended to serve data older than a year. The maximum time range in which aggregates can be served may depend on two factors. One factor may be that the 10 MB limit is breached and the second factor may be that latency issues start to occur because of extremely large record sizes. The top 24 largest record sizes in one implementation are shown in illustration 414 of Figure 4C, which shows a maximum record size of 484 kilobytes. Accordingly, in at least this implementation, the algorithm can scale well as there is still up to 9.5 MB available before the limit can be breached in a worst-case scenario.

[33] A TimescaleDB™ database may be used as a second database for its performance advantage involving calculations over time series data. In TimescaleDB™,

the data is stored in an abstraction called chunks, where each chunk contains some data within a specific time range. These chunks help to narrow down the amount of data that needs to be queried at a time. It also creates and stores all the indexes for each chunk separately. It will be appreciated that “chunk” and “segment” may be used interchangeably in the present disclosure.

[34] In at least some embodiments, a user may be any suitable type of entity, which may include a person, a consumer looking to purchase a product or service via a transaction processing server, a seller or merchant looking to sell a product or service via the transaction processing server, a motorcycle driver or pillion rider in a case of the user looking to book or provide a motorcycle ride via the transaction processing server, a car driver or passenger in a case of the user looking to book or provide a car ride via the transaction processing server, and other similar entity. A user who is registered to the transaction processing or data processing server will be called a registered user. A user who is not registered to the transaction processing server or data processing server will be called a non-registered user. The term user will be used to collectively refer to both registered and non-registered users. A user may interchangeably be referred to as a requestor (e.g. a person who requests for a product or service) or a provider (e.g. a person who provides the requested product or service to the requestor).

[35] In at least some embodiments, a data processing server is a server that hosts software application programs for performing data processing in relation to a request for data. The data processing server may be implemented as shown in schematic diagram 300 of Fig. 3 for adaptively processing a request for data.

[36] In at least some embodiments, a transaction processing server is a server that hosts software application programs for processing payment transactions for, for example, purchasing of a good or service by a user. The transaction processing server communicates with any other servers (e.g., a data processing server) concerning processing payment transactions relating to the purchasing of the good or service. For example, data relating to an approved or rejected transaction (e.g. date, time, amount, currency, user name, and other similar data relating to the concerned transaction) may be provided to the data processing server and processed into aggregated data. The aggregated data may then be stored or transferred to a database. In an implementation, the transaction processing server may also be in communication with a database directly

which will store the data relating to an approved or rejected transaction as raw data, or may also be configured to process the data into aggregated data for a time range before doing so. In a case where processing data in relation to a request for data is a service provided by the platform, the transaction processing server may communicate with the data processing server to facilitate payment for the data processing service after data relating to a request for data is retrieved and provided to the requestor. The transaction processing server may use a variety of different protocols and procedures in order to process the payment and/or travel co-ordination requests.

[37] Transactions that may be performed via a transaction processing server include product or service purchases, credit purchases, debit transactions, fund transfers, account withdrawals, etc. Transaction processing servers may be configured to process transactions via cash-substitutes, which may include payment cards, letters of credit, checks, payment accounts, etc.

[38] In at least some embodiments, the transaction processing server is usually managed by a service provider that may be an entity (e.g. a company or organization) which operates to process transaction requests and/or travel co-ordination requests e.g. pair a provider of a travel co-ordination request to a requestor of the travel co-ordination request. The transaction processing server may include one or more computing devices that are used for processing transaction requests and/or travel co-ordination requests.

[39] In at least some embodiments, a transaction account is an account of a user who is registered at a transaction processing server. The user can be a customer, a merchant providing a product for sale on a platform and/or for onboarding the platform, a hail provider (e.g., a driver), or any third parties (e.g., a courier) who want to use the transaction processing server. In certain circumstances, the transaction account is not required to use the transaction processing server. A transaction account includes details (e.g., name, address, vehicle, face image, etc.) of a user. The transaction processing server manages the transaction.

[40] Embodiments will be described, by way of example only, with reference to the drawings. Like reference numerals and characters in the drawings refer to like elements or equivalents.

[41] Some portions of the description which follows are explicitly or implicitly presented in terms of algorithms and functional or symbolic representations of operations on data within a computer memory. These algorithmic descriptions and functional or symbolic representations are the means used by those skilled in the data processing arts to convey most effectively the substance of their work to others skilled in the art. An algorithm is here, and generally, conceived to be a self-consistent sequence of steps leading to a desired result. The steps are those requiring physical manipulations of physical quantities, such as electrical, magnetic or optical signals capable of being stored, transferred, combined, compared, and otherwise manipulated.

[42] Unless specifically stated otherwise, and as apparent from the following, it will be appreciated that throughout the present specification, discussions utilizing terms such as “comparing”, “determining”, “calculating”, “retrieving”, “processing”, “storing”, “aggregating”, “subtracting”, “adding”, or the like, refer to the action and processes of a computer system, or similar electronic device, that manipulates and transforms data represented as physical quantities within the computer system into other data similarly represented as physical quantities within the computer system or other information storage, transmission or display devices.

[43] In addition, the present specification also implicitly discloses a computer program, in that it would be apparent to the person skilled in the art that the individual steps of the method described herein may be put into effect by computer code. The computer program is not intended to be limited to any particular programming language and implementation thereof. It will be appreciated that a variety of programming languages and coding thereof may be used to implement the teachings of the disclosure contained herein. Moreover, the computer program is not intended to be limited to any particular control flow. There are many other variants of the computer program, which can use different control flows without departing from the scope of the specification.

[44] Furthermore, one or more of the steps of the computer program may be performed in parallel rather than sequentially. Such a computer program may be stored on any computer readable medium. The computer readable medium may include storage devices such as magnetic or optical disks, memory chips, or other storage devices suitable for interfacing with a computer. The computer readable medium may also include a hard-wired medium such as exemplified in the Internet system, or wireless medium such as

exemplified in the GSM mobile telephone system. The computer program when loaded and executed on such a computer effectively results in an apparatus that implements the steps of the preferred method.

[45] Fig. 1 illustrates a block diagram of an example system 100 for adaptively processing a request for data. In some embodiments, the system 100 enables a payment transaction for a good or service, and/or a request for a ride or delivery of a physical item (e.g. one or more food items or a parcel) between a requestor and a provider.

[46] The system 100 comprises a requestor device 102, a provider device 104, an acquirer server 106, a transaction processing server 108, an issuer server 110, a data processing server 140 and two reference databases 150A and 150B.

[47] The requestor device 102 is in communication with a provider device 104 via a connection 112. The connection 112 may be wireless (e.g., via NFC communication, Bluetooth, etc.) or over a network (e.g., the Internet). The requestor device 102 is also in communication with the data processing server 140 via a connection 121. The connection 121 may be via a network (e.g., the Internet). The requestor device 102 may also be connected to a cloud that facilitates the system 100 for adaptively processing a request for data. For example, the requestor device 102 can send a signal or data to the cloud directly via a wireless connection (e.g., via NFC communication, Bluetooth, etc.) or over a network (e.g., the Internet).

[48] The provider device 104 is in communication with the requestor device 102 as described above, usually via the transaction processing server 108. The provider device 104 is, in turn, in communication with an acquirer server 106 via a connection 114. The provider device 104 is also in communication with the data processing server 140 via a connection 123. The connections 114 and 123 may be via a network (e.g., the Internet). The provider device 104 may also be connected to a cloud that facilitates the system 100 for adaptively processing a request for data. For example, the provider device 104 can send a signal or data to the cloud directly via a wireless connection (e.g., via NFC communication, Bluetooth, etc.) or over a network (e.g., the Internet).

[49] The acquirer server 106, in turn, is in communication with the transaction processing server 108 via a connection 116. The transaction processing server 108, in

turn, is in communication with an issuer server 110 via a connection 118. The connections 116 and 118 may be via a network (e.g., the Internet).

[50] The transaction processing server 108 is further in communication with the data processing server 140 via a connection 120. The connection 120 may be over a network (e.g., a local area network, a wide area network, the Internet, etc.). In one arrangement, the transaction processing server 108 and the data processing server 140 are combined and the connection 120 may be an interconnected bus.

[51] The data processing server 140, in turn, is in communication with the reference databases 150A and 150B via respective connection 122. The connection 122 may be a network (e.g., the Internet). The data processing server 140 may also be connected to a cloud that facilitates the system 100 for adaptively processing a request for data. For example, the data processing server 140 can send a signal or data to the cloud directly via a wireless connection (e.g., via NFC communication, Bluetooth, etc.) or over a network (e.g., the Internet).

[52] The reference databases 150A and 150B may comprise data relating users, transactions, products, services, and other similar data, for example relating to a concerned platform. The data may be raw data that is compiled over a predetermined time range, or aggregated data relating to a predetermined time range. For example, reference database 150A may be NoSQL database storing aggregated data relating to a predetermined time range, while reference database 150B may be a time series database storing raw data that is compiled over a predetermined time range. The reference databases 150A and 150B may be combined with the data processing server 140. In an example, the reference databases 150A and 150B may be managed by an external entity and the data processing server 140 is a server that, based on a start time and end time indicated in a request for data (e.g. the request being provided by a user, or provided by staff of the concerned platform, for example by a risk management team personnel for the platform, or other similar users), compares a first time range with a second time range, the first time range being based on the start time and end time indicated in the request for data associated with the start time and the end time, and the second time range being associated with reference database 150A or reference database 150B. The data processing server 140 may then determine one or more sub time ranges for retrieving the data, the one or more sub time ranges being based on the first time range and the

comparison. Further, the data processing server 140 may then determine whether to retrieve data associated with each of the one or more sub time ranges from the reference database 150A or 150B based on the comparison. In an implementation, there may be more than two reference databases, in which the data processing server 140 may be configured to determine which database to use based on the request for data. The reference databases 150A and 150B (including any additional reference databases as explained above) are collectively referred to herein as reference databases 150.

[53] Alternatively, one or more modules may store the raw data or aggregated data instead of the reference databases 150A or 150B, wherein the module may be integrated as part of the data processing server 140 or external from the data processing server 140.

[54] In the illustrative embodiment, each of the devices 102, 104, and the servers 106, 108, 110, 140, and/or reference databases 150 provides an interface to enable communication with other connected devices 102, 104 and/or servers 106, 108, 110, 140, and/or reference databases 150. Such communication is facilitated by an application programming interface ("API"). Such APIs may be part of a user interface that may include graphical user interfaces (GUIs), Web-based interfaces, programmatic interfaces such as application programming interfaces (APIs) and/or sets of remote procedure calls (RPCs) corresponding to interface elements, messaging interfaces in which the interface elements correspond to messages of a communication protocol, and/or suitable combinations thereof. For example, it is possible for at least one of the requestor device 102 and the provider device 104 to send data relating to a product in response to an enquiry shown on the GUI running on the respective API.

[55] Use of the term 'server' herein can mean a single computing device or a plurality of interconnected computing devices which operate together to perform a particular function. That is, the server may be contained within a single hardware unit or be distributed among several or many different hardware units.

[56] The data processing server 140 is associated with an entity (e.g. a company or organization or moderator of the service). In one arrangement, the data processing server 140 is owned and operated by the entity operating the transaction processing server 108. In such an arrangement, the data processing server 140 may be implemented as a part

(e.g., a computer program module, a computing device, etc.) of the transaction processing server 108.

[57] The transaction processing server 108 may also be configured to manage the registration of users. A registered user has a transaction account (see the discussion above) which includes details of the user. The registration step is called on-boarding. A user may use either the requestor device 102 or the provider device 104 to perform on-boarding to the transaction processing server 108.

[58] It may not be necessary to have a transaction account at the transaction processing server 108 to access the functionalities of the transaction processing server 108. However, there are functions that are available to a registered user. These additional functions will be discussed below.

[59] The on-boarding process for a user is performed by the user through one of the requestor device 102 or the provider device 104. In one arrangement, the user downloads an app (which includes the API to interact with the transaction processing server 108) to the requestor device 102 or the provider device 104. In another arrangement, the user accesses a website (which includes the API to interact with the transaction processing server 108) on the requestor device 102 or the provider device 104. The user is then able to interact with the data processing server 140. The user may be a requestor or a provider associated with the requestor device 102 or the provider device 104, respectively.

[60] Details of the registration may include, for example, name of the user, address of the user, emergency contact, blood type or other healthcare information, next-of-kin contact, permissions to retrieve data and information from the requestor device 102 and/or the provider device 104 for product identification purposes, such as permission to use a camera of the requestor device 102 and/or the provider device 104 to take a picture of the user for identification purposes. Alternatively, another mobile device may be selected instead of the requestor device 102 and/or the provider device 104 for retrieving the data. Once on-boarded, the user would have a transaction account that stores all the details.

[61] The requestor device 102 is associated with a customer (or requestor) who is a party to a transaction that occurs between the requestor device 102 and the provider device 104, or between the requestor device 102 and the data processing server 140.

The requestor device 102 may be a computing device such as a desktop computer, an interactive voice response (IVR) system, a smartphone, a laptop computer, a personal digital assistant computer (PDA), a mobile computer, a tablet computer, and the like.

[62] The requestor device 102 includes transaction credentials (e.g., a payment account) of a requestor to enable the requestor device 102 to be a party to a payment transaction. If the requestor has a transaction account, the transaction account may also be included (i.e., stored) in the requestor device 102. For example, a mobile device (which is a requestor device 102) may have the transaction account of the customer stored in the mobile device.

[63] In one example arrangement, the requestor device 102 is a computing device in a watch or similar wearable and is fitted with a wireless communications interface (e.g., a NFC interface). The requestor device 102 can then electronically communicate with the provider device 104 regarding a transaction request. The customer uses the watch or similar wearable to make request regarding the transaction request by pressing a button on the watch or wearable.

[64] The provider device 104 is associated with a provider who is also a party to the transaction request that occurs between the requestor device 102 and the provider device 104. The provider device 104 may be a computing device such as a desktop computer, an interactive voice response (IVR) system, a smartphone, a laptop computer, a personal digital assistant computer (PDA), a mobile computer, a tablet computer, and the like.

[65] Hereinafter, the term “provider” refers to a service provider and any third party associated with providing a product or service for purchase, or a travel or ride or delivery service via the provider device 104. Therefore, the transaction account of a provider refers to both the transaction account of a provider and the transaction account of a third party (e.g., a travel co-ordinator or merchant) associated with the provider.

[66] If the provider has a transaction account, the transaction account may also be included (i.e., stored) in the provider device 104. For example, a mobile device (which is a provider device 104) may have the transaction account of the provider stored in the mobile device.

[67] In one example arrangement, the provider device 104 is a computing device in a watch or similar wearable and is fitted with a wireless communications interface (e.g., a NFC interface). The provider device 104 can then electronically communicate with the requestor to make request regarding the transaction request by pressing a button on the watch or wearable.

[68] The acquirer server 106 is associated with an acquirer who may be an entity (e.g. a company or organization) which issues (e.g. establishes, manages, administers) a payment account (e.g. a financial bank account) of a merchant. Examples of the acquirer include a bank and/or other financial institution. As discussed above, the acquirer server 106 may include one or more computing devices that are used to establish communication with another server (e.g., the transaction processing server 108) by exchanging messages with and/or passing information to the other server. The acquirer server 106 forwards the payment transaction relating to a transaction request to the transaction processing server 108.

[69] The transaction processing server 108 is configured to process processes relating to a transaction account by, for example, forwarding data and information associated with the transaction to the other servers in the system 100 such as the data processing server 140. In an example, the transaction processing server 108 may transmit data relating to an approved or rejected transaction (e.g. date, time, amount, currency, user name, and other similar data relating to the concerned transaction) to the data processing server 140. In a case where processing data in relation to a request for data is a service provided by the platform, the transaction processing server 108 may communicate with the data processing server 140 to facilitate payment for the data processing service after data relating to a request for data is retrieved and provided to the requestor. The transaction processing server 108 may use a variety of different protocols and procedures in order to process the payment and/or travel co-ordination requests. It will be appreciated that payment for a transaction may be made via a variety of methods such as credit cards, debit cards, digital wallets, buy-first pay-later schemes, and other similar payment methods.

[70] The issuer server 110 is associated with an issuer and may include one or more computing devices that are used to perform a payment transaction. The issuer may be an entity (e.g. a company or organization) which issues (e.g. establishes, manages,

administers) a transaction credential or a payment account (e.g. a financial bank account) associated with the owner of the requestor device 102. As discussed above, the issuer server 110 may include one or more computing devices that are used to establish communication with another server (e.g., the transaction processing server 108) by exchanging messages with and/or passing information to the other server.

[71] Each of the reference database 150A and 150B is a database or server associated with an entity (e.g. a company or organization) which manages (e.g. establishes, administers) data relating to users, transactions, products, services, and other similar data, for example relating to the entity. In an arrangement, the reference database 150A may be a first database that stores data for an amount of time before the data is purged or deleted from the database. The data stored in the reference database 150A may comprise a plurality of datapoints, each datapoint comprising information that is aggregated for a predetermined time range. There may be one or more different amounts of time that may be set. For example, a first amount of time may be set at 30 days for storing data that is aggregated for a predetermined time range of one day (e.g. for daily aggregates), and a second amount of time may be set at 12 months for storing data that is aggregated for a predetermined time range of one month (e.g. for monthly aggregates). It will be appreciated that other amounts of time may be set for other predetermined time ranges for data that is aggregated on weekly, yearly, or other periodic basis. Further, the reference database 150B may be a second database that stores data in one or more segments, each segment comprising raw data that is compiled within a predetermined time range.

[72] Advantageously, the system 100 enables data to be efficiently retrieved such that most of the data can be obtained from pre-calculated aggregates and the least amount of data has to be fetched by doing operations on raw data, reducing latency issues.

[73] Fig. 2 illustrates a schematic diagram of an example data processing server 140 according to various embodiments. The data processing server 140 may comprise a data module 260 configured to receive data and information from the requestor device 102, provider device 104, transaction processing server 108, reference databases 150, a cloud and other sources of information to facilitate processing a request for data by the data processing server 140. For example, the data module 260 may be configured to receive data and information required for processing the request for data from the requestor device

102, the provider device 104, transaction processing server 108, reference databases 150, and/or other sources of information. The data module 260 may also be configured to retrieve data required by the request from the reference databases 150 including reference databases 150A and 150B, other databases that may be designated by the request for data, and/or other sources of information. The data module 260 may be further configured to send information relating to data retrieved in response to the request for data to the requestor device 102, the provider device 104, the transaction processing server 108, or other destinations where the information is required.

[74] The data processing server 140 may comprise a comparison module 262 that is configured for comparing a first time range with a second time range, the first time range being based on a start time and an end time indicated in a request for data associated with the start time and the end time, and the second time range being associated with a first database (e.g. the reference database 150A) or a second database (e.g. the reference database 150B).

[75] The data processing server 140 may also comprise a determination module 264 that is configured for determining one or more sub time ranges for retrieving the data, the one or more sub time ranges being based on the first time range and the comparison by the comparison module 262. The determination module 264 is further configured for assessing whether to retrieve data associated with each of the one or more sub time ranges from the first database or the second database based on the comparison. The determination of the first and second time ranges, as well as the one or more sub time ranges is further explained in flowchart 600 of Figs. 6A – 6E. It will be appreciated that the first and second time ranges, as well as the one or more sub time ranges that are determined in some of the steps in the flowchart 600 may vary from one another. Note that in Figs. 6A-6E, “original start time” and “original end time” respectively refer to the start time and end time specified in the data request.

[76] The data processing server 140 may also comprise a calculation module 266 that is configured for subtracting or adding data retrieved from the first and second databases. The data processing server 140 may also comprise an aggregation module 268 that is configured for processing the raw data retrieved from the second database into one or more datapoints comprising information that is aggregated for the concerned time range.

In an implementation, aggregation on raw data (e.g. retrieved from the second database) may be performed by operators in SQL instead of the aggregation module 268.

[77] In an implementation, the second database stores data in one or more segments, each segment comprising data within a predetermined time range. The first time range starts from the start time to a current time at which the comparison is made, and the second time range is the predetermined time range of the second database. Determining the one or more sub time ranges may comprise determining a sub time range that starts from the start time to the end time if the first time range is shorter than the second time range. Further, determining whether to retrieve data associated with each of the one or more sub time ranges from the first database or the second database further comprises determining that the data associated with the sub time range is to be retrieved from the second database.

[78] In an implementation, wherein the second database stores data in one or more segments, each segment comprising data within a predetermined time range, wherein the first time range starts from the start time to a current time at which the comparison is made and is shorter than the predetermined time range of the second database, and the second time range is an amount of time in which data is kept in the first database before being deleted from the first database, determining the one or more sub time ranges further comprises determining whether the first-time range is shorter than the second time range.

[79] In an implementation, wherein the first time range starts from the start time to a current time at which the comparison is made and the second time range is a first amount of time in which data is kept in the first database before being deleted from the first database, comparing the first time range with the second time range by the comparison module 262 further comprises: comparing the first time range with a third time range if the first time range is longer than the second time range, wherein the third time range is a second amount of time in which data is kept in the first database before being deleted from the first database, the third time range being longer than the second time range; and determining, by the determination module 264, the one or more sub time ranges based on the comparison.

[80] In an implementation, determining the one or more sub time ranges by the determination module 264 further comprises determining a first sub time range and a

second sub time range, the first time range being an addition of the first sub time range with the second sub time range. Data associated with each of the first and second sub time range are retrieved by the data module 260 from the first database or second database based on the comparison, and the retrieved data are added together by the calculation module 268.

[81] In an implementation, determining the one or more sub time ranges by the determination module 264 further comprises determining a first sub time range and a second sub time range, the first time range being a subtraction of the second sub time range from the first sub time range. Data associated with the first sub time range is retrieved by the data module 260 from the first database, and retrieving data associated with the second sub time range by the data module 260 from the second database based on the comparison. The data retrieved from the second database is subtracted by the calculation module 268 from the data retrieved from the first database.

[82] In an implementation, wherein the first database stores data comprising a plurality of datapoints, each datapoint comprising information that is aggregated for a predetermined time range, and the second database stores raw data, retrieving data associated with the first sub time range from the first database comprises retrieving a plurality of datapoints associated with the first sub time range from the first database, and retrieving data associated with the second sub time range from the second database comprises retrieving raw data associated with the second sub time range from the second database. Subtracting the data retrieved from the second database from the data retrieved from the first database further comprises processing, by the aggregation module 268, the raw data retrieved from the second database into one or more datapoints comprising information that is aggregated for the second sub time range, and subtracting, by the calculation module 266, the one or more datapoints processed from raw data retrieved from the second database from the plurality of datapoints associated with the first sub time range.

[83] Each of the data module 260, comparison module 262, determination module 264, calculation module 266 and aggregation module 268 may further be in communication with a processing module (not shown) of the data processing server 140, for example for coordination of respective tasks and functions during the process. The data module 260 may be further configured to communicate with and store data and information for each of

the processing module, comparison module 262, determination module 264, calculation module 266 and aggregation module 268. Alternatively, all the tasks and functions required for adaptively processing a request for data may be performed by a single processor of the data processing server 140.

[84] A core component of at least some embodiments of the algorithm for adaptively processing a request for data is to build an expression tree such as shown in illustration 500 of Fig. 5, and evaluate it as efficiently as possible. In the example of Fig. 5, the tree consists of three kinds of nodes. AddOp nodes 502 and 504 perform addition operations, and are configured to add up all the data that are obtained from their child nodes. For example, AddOp node 504 may be configured to add up data that are collected from nodes 510, 512 and 514, and AddOp node 502 may be configured to add up data that are collected from nodes 504, 506 and 508. SubOp node 506 performs subtraction operations, and is configured to subtract data obtained from, for example, a right child node from a left child node. A SubOp node thus has only two child nodes. For example, SubOp node 506 may be configured to subtract data collected by node 518 from data collected by node 516. Further, DataOp nodes 508, 510, 512, 514, 516 and 518 perform data operations and are the terminal nodes in the expression tree. These nodes are configured to, based on outcome of the processing of the request for data, call the relevant database (e.g. reference databases 150A or 150B) and fetch the required data accordingly. It will be appreciated that each of the nodes 508, 510, 512, 514, 516 and 518 may be processed independently or in parallel with one another.

[85] During implementation of the expression tree of illustration 500, the algorithm first tries to determine if a given time range can be retrieved directly from a single segment of data in a second database, for example a TimescaleDB™ database. If not, then the algorithm moves on to see if most of the data can be fetched from daily aggregated data stored in a first database, for example an Aerospike™ database. Data for residual time windows (e.g. sub time ranges for retrieving data) that are left at the start and end of the requested data are fetched from the raw data kept in the second database. For example, if today's time is 2021-10-01 00:00:00 (e.g. a current time) and data from 2021-08-31 16:00:00 (e.g. a start time indicated in a request for data) till 2021-09-30 16:00:00 (e.g. an end time indicated in the request for data) is required, then data from 2021-09-01 00:00:00 till 2021-09-29 23:59:59 will be fetched from daily aggregates stored in the first database. Data from 2021-08-31 16:00:00 till 2021-08-31 23:59:59 and from 2021-09-30 00:00:00

till 2021-09-30 16:00:00 will be fetched from the raw data stored in the second database. Finally, the algorithm moves on to see if most of the data can be fetched from monthly aggregates stored in the first database. Data from any residual time windows that are left at the start and end are either fetched directly from the raw data in the second database, or if that is not optimal (e.g. too many segments have to be parsed to fetch the data), the monthly aggregate from the second database is utlized and the overlapping time range portion between the data retrieved from the first and second databases is subtracted to arrive at the final value. For example, if today's time is 2021-10-01 00:00:00 and data from 2021-06-03 00:00:00 till 2021-07-03 00:00:00 is requested in the request for data, the algorithm may fetch raw data from the second database in between 2021-07-01 00:00:00 and 2021-07-03 00:00:00. Since the whole of the data for June cannot be fetched from the second database, the time range may be fragmented such that it is most optimal to fetch aggregates from the raw data stored in the second database in between 2021-06-01 00:00:00 and 2021-06-02 23:59:59 and subtract that from the monthly aggregate stored in the first database for June.

[86] An example of a detailed expression tree creation logic is represented as a flow chart 600 in Figures 6A – 6E. The process begins at step 602. At step 604, a start time and an end time indicated in a request for data (e.g. a request for data associated with the start time and the end time) is obtained. At step 606, it is determined whether a time range that starts from the start time to a current time (e.g. a current time at which the comparison is made) is shorter than a segment size of the second database (e.g. the reference database 150B), or whether a time range that starts from the start time to the end time is shorter than the segment size. If it is determined to be the case, the process proceeds to step 608 wherein a data collection operation node is created which performs the operation of retrieving data associated with the start time and the end time from the second database. Otherwise, the process proceeds to step 610 where it is determined whether a time range starting from the start time to the current time is shorter than a Day Purge Duration (e.g. a first amount of time in which data is stored in the first database (e.g. the reference database 150A) before the stored data is deleted from the first database. In the present example, the Day Purge Duration is set as 30 days.). If it is determined to be the case, the process proceeds to step 612 where an addition operation node is created. The process proceeds to steps 614 and 616.

[87] At step 614, it is determined if the current time is the same as the end time, or whether the end time is the same as the same day's end of day (e.g. whether the end time is 23:59:59 of the end time's date). If it is determined to be the case, the process proceeds to step 618 where a 'tempEndTime' is defined as the end time. Otherwise, the process proceeds to step 620 where a data collection operation node is created which performs the operation of retrieving data associated with a time range (e.g. a first sub time range for the addition operation of step 612) that starts from 00:00:00 of the end time's date to the end time from the second database. Further, a 'tempEndTime' is defined as 23:59:59 of the end time's previous day. On the other hand, at step 616, it is determined whether the start time is not the same as the same day's start of day (e.g. whether the start time is not 00:00:00 of the start time's date). If it is determined that the start time is not 00:00:00 of the start time's date, the process proceeds to step 622 where a data collection operation node is created which performs the operation of retrieving data associated with a time range (e.g. a second sub time range for the addition operation of step 612) that starts from the start time to the start time's end of day from the second database. Further, a 'tempStartTime' is defined as 00:00:00 of a next day of the start time. Otherwise, the process proceeds to step 624 wherein a 'tempStartTime' is defined as the start time. From steps 618 and 624, the process proceeds to step 626 where it is determined whether the 'tempStartTime' is earlier than the 'tempEndTime'. If it is determined to be the case, the process proceeds to step 628 where a data collection operation node is created which performs the operation of retrieving data associated with a time range (e.g. a third sub time range for the addition operation of step 612) from the 'tempStartTime' to 'tempEndTime' (as respectively defined in steps 624 and 618) from the first database. The data collected from the node created in step 628 would be the required data in response to the request for data.

[88] On the other hand, from steps 620 and 622, the process also proceeds to step 626 where it is determined whether the 'tempStartTime' is earlier than the 'tempEndTime'. If it is determined to be the case, the process proceeds to step 628 where a data collection operation node is created which performs the operation of retrieving data associated with a time range from the 'tempStartTime' to 'tempEndTime' (as respectively defined in steps 622 and 620) from the first database. The data obtained from the nodes created in steps 620 (e.g. data corresponding to the first sub time range for the addition operation of step 612), 622 (e.g. data corresponding to the second sub time range for the addition operation of step 612) and 628 (e.g. data corresponding to the third sub time range for the addition

operation of step 612, if any) are then added together by the addition operation node created in step 612 to obtain the required data in response to the request for data.

[89] On the other hand, if it is determined in step 610 that the time range from the start time to the current time is longer than the Day Purge Duration of the first database, the process proceeds to step 630 where it is determined whether a time range from the start time to the current time is shorter than a Month Purge Duration (e.g. a second amount of time in which data is stored in the first database (e.g. the reference database 150A) before the stored data is deleted from the first database, wherein the second amount of time is longer than the first amount of time. In the present example, the Month Purge Duration is set as 12 months.). If it is determined to be the case, the process proceeds to step 632 where an add operation node is created. The process then proceeds to steps 634 and 636. In step 634, it is determined whether the end time is the same as the current time, or whether the end time is 23.59:59 of the last day of the month of the end time. If it is determined to be the case, the process proceeds to step 638 where a 'tempEndTime' is defined as the end time. Otherwise, the process proceeds to step 640 where a 'tempEndTime' is defined as 23:59:59 of the last day of the previous month of the end time. In step 636, it is determined whether the start time is 00:00:00 in the first day of the month of the start time. If it is determined to be the case, the process proceeds to step 644 where a 'tempStartTime' is defined as the start time. Otherwise, the process proceeds to step 642 where the 'tempStartTime' is defined as 00:00:00 of the first day of the next month of the start time. From step 638 or 640 and step 642 or 644, the respectively defined 'tempEndTime' and 'tempStartTime' are used in step 646 to determine whether the 'tempStartTime' is earlier than the 'tempEndTime'. If it is determined to be the case, a data collection operation node is created at step 648 which performs the operation of retrieving data associated with the time range between the 'tempStartTime' and 'tempEndTime' from the first database.

[90] Further, from step 640, the process also proceeds to step 650 where it is determined whether a time range starting from the end time's start of month to the end time is lesser than or equal to 15 days. If it is determined to be the case, the process proceeds to step 652 where a data collection operation node is created which performs the operation of retrieving data associated with a time range starting from 00:00:00 of the first day of the month of the end time to the end time from the second database. Otherwise, the process proceeds to step 654 where a subtraction operation node is created. The

process further proceeds to step 656 where a data collection operation node is created (e.g. a first child node of the subtraction operation node of step 654) which performs the operation of retrieving data associated with a time range (e.g. a first sub time range for the subtraction operation of step 654) starting from 00:00:00 of the first day of the month of the end time to 23:59:59 of the last day of the month of the end time from the first database. The process further proceeds to step 658 where another data collection operation node is created (e.g. a second child node of the subtraction operation node of step 654) which performs the operation of retrieving data associated with a time range (e.g. a second sub time range for the subtraction operation of step 654) starting from the end time to 23:59:59 of the last day of the month of the end time from the second database. Accordingly, the subtraction operation node of step 654 is configured to subtract data retrieved by the second child node (e.g. data associated with the second sub time range for the subtraction operation of step 654) from data retrieved by the first child node (e.g. data associated with the first sub time range for the subtraction operation of step 654). Before the subtraction, raw data retrieved from the second database may be processed (e.g. by the aggregation module 268) into data that is aggregated for the time range starting from 00:00:00 of the first day of the month of the end time to 23:59:59 of the last day of the month of the end time.

[91] Further, from step 642, the process also proceeds to step 660 where it is determined whether a time range starting from the start time to the start time's end of month is lesser than or equal to 15 days. If it is determined to be the case, the process proceeds to step 662 where a data collection operation node is created which performs the operation of retrieving data associated with a time range starting from the start time to 23:59:59 of the last day of the month of the start time from the second database. Otherwise, the process proceeds to step 664 where a subtraction operation node is created. The process further proceeds to step 666 where a data collection operation node is created (e.g. a first child node of the subtraction operation node of step 664) which performs the operation of retrieving data associated with a time range (e.g. a first sub time range for the subtraction operation of step 664) starting from 00:00:00 of the first day of the month of the start time to 23:59:59 of the last day of the month of the start time from the first database. The process further proceeds to step 668 where another data collection operation node is created (e.g. a second child node of the subtraction operation node of step 664) which performs the operation of retrieving data associated with a time range (e.g. a second sub time range for the subtraction operation of step 664) starting from

00:00:00 of the first day of the month of the start time to the start time from the second database. Accordingly, the subtraction operation node of step 664 is configured to subtract data retrieved by the second child node (e.g. data associated with the second sub time range for the subtraction operation of step 654) from data retrieved by the first child node (e.g. data associated with the first sub time range for the subtraction operation of step 654). Before the subtraction, raw data retrieved from the second database may be processed (e.g. by the aggregation module 268) into data that is aggregated for the time range starting from 00:00:00 of the first day of the month of the start time to the start time.

[92] Accordingly, the addition operation node of step 632 is configured to add up all the data collected (if any) from its child nodes, for example, the nodes created in steps 652, 654, 648, 662 and 664, to obtain the required data in response to the request for data. It will be appreciated that the Day Purge Duration and Month Purge Duration may vary depending on application, and other purge durations may also be implemented, such as a Weekly Purge Duration for weekly aggregates. For example, the Day Purge Duration may be set to 7 days instead of 30 days for daily aggregates, and/or a further Week Purge Duration may be set as 30 days for weekly aggregates.

[93] Fig. 7 illustrates an example flow diagram of a method for adaptively processing a request for data according to various embodiments. In a step 702, a first time range is compared with a second time range, the first time range being based on a start time and an end time indicated in a request for data associated with the start time and the end time, and the second time range being associated with a first database or a second database. In a step 704, one or more sub time ranges are determined for retrieving the data, the one or more sub time ranges being based on the first time range and the comparison. In a step 706, it is determined whether to retrieve data associated with each of the one or more sub time ranges from the first database or the second database based on the comparison.

[94] Fig. 8A depict an example computer system 1400, in accordance with which the data processing server 140 described can be practiced. The computer system 1400 includes a computer module 1401. An external Modulator-Demodulator (Modem) transceiver device 1416 may be used by the computer module 1401 for communicating to and from a communications network 1420 via a connection 1421. The communications network 1420 may be a wide-area network (WAN), such as the Internet, a cellular telecommunications network, or a private WAN. Where the connection 1421 is a

telephone line, the modem 1416 may be a traditional “dial-up” modem. Alternatively, where the connection 1421 is a high capacity (e.g., cable) connection, the modem 1416 may be a broadband modem. A wireless modem may also be used for wireless connection to the communications network 1420.

[95] The computer module 1401 typically includes at least one processor unit 1405, and a memory unit 1406. For example, the memory unit 1406 may have semiconductor random access memory (RAM) and semiconductor read only memory (ROM). The computer module 1401 also includes an interface 1408 for the external modem 1416. In some implementations, the modem 1416 may be incorporated within the computer module 1401, for example within the interface 1408. The computer module 1401 also has a local network interface 1411, which permits coupling of the computer system 1400 via a connection 1423 to a local-area communications network 1422, known as a Local Area Network (LAN). As illustrated in Fig. 8A, the local communications network 1422 may also couple to the wide network 1420 via a connection 1424, which would typically include a so-called “firewall” device or device of similar functionality. The local network interface 1411 may comprise an Ethernet circuit card, a Bluetooth® wireless arrangement or an IEEE 802.11 wireless arrangement; however, numerous other types of interfaces may be practiced for the interface 1411.

[96] The I/O interfaces 1408 may afford either or both of serial and parallel connectivity, the former typically being implemented according to the Universal Serial Bus (USB) standards and having corresponding USB connectors (not illustrated). Storage devices 1409 are provided and typically include a hard disk drive (HDD) 1410. Other storage devices such as a floppy disk drive and a magnetic tape drive (not illustrated) may also be used. An optical disk drive 1412 is typically provided to act as a non-volatile source of data. Portable memory devices, such optical disks, USB-RAM, portable, external hard drives, and floppy disks, for example, may be used as appropriate sources of data to the system 1400.

[97] The components 1405 to 1412 of the computer module 1401 typically communicate via an interconnected bus 1304 and in a manner that results in a conventional mode of operation of the computer system 1400 known to those in the relevant art. For example, the processor 1405 is coupled to the system bus 1404 using a connection 1418. Likewise, the memory 1406 and optical disk drive 1412 are coupled to

the system bus 1404 by connections 1419. Examples of computers on which the described arrangements can be practised include IBM-PC's and compatibles, Sun Sparcstations, Apple or like computer systems.

[98] The method 700, where performed by the data processing server 140 may be implemented using the computer system 1400. The processes may be implemented as one or more software application programs 1433 executable within the computer system 1400. In particular, the sub-processes 400, 500, and 600 are effected by instructions in the software 1433 that are carried out within the computer system 1400. The software instructions may be formed as one or more code modules, each for performing one or more particular tasks. The software may also be divided into two separate parts, in which a first part and the corresponding code modules performs the methods and a second part and the corresponding code modules manage a user interface between the first part and the user.

[99] The software may be stored in a computer readable medium, including the storage devices described below, for example. The software is loaded into the computer system 1400 from the computer readable medium, and then executed by the computer system 1400. A computer readable medium having such software or computer program recorded on the computer readable medium is a computer program product. The use of the computer program product in the computer system 1400 preferably effects an advantageous apparatus for a data processing server 140.

[100] The software 1433 is typically stored in the HDD 1410 or the memory 1406. The software is loaded into the computer system 1400 from a computer readable medium, and executed by the computer system 1400. Thus, for example, the software 1433 may be stored on an optically readable disk storage medium (e.g., CD-ROM) 1425 that is read by the optical disk drive 1412. A computer readable medium having such software or computer program recorded on it is a computer program product. The use of the computer program product in the computer system 1400 preferably effects an apparatus for a data processing server 140.

[101] In some instances, the application programs 1433 may be supplied to the user encoded on one or more CD-ROMs 1425 and read via the corresponding drive 1412, or alternatively may be read by the user from the networks 1420 or 1422. Still further, the

software can also be loaded into the computer system 1400 from other computer readable media. Computer readable storage media refers to any non-transitory tangible storage medium that provides recorded instructions and/or data to the computer system 1400 for execution and/or processing. Examples of such storage media include floppy disks, magnetic tape, optical disc, a hard disk drive, a ROM or integrated circuit, USB memory, a magneto-optical disk, or a computer readable card such as a PCMCIA card and the like, whether or not such devices are internal or external of the computer module 1401. Examples of transitory or non-tangible computer readable transmission media that may also participate in the provision of software, application programs, instructions and/or data to the computer module 1401 include radio or infra-red transmission channels as well as a network connection to another computer or networked device, and the Internet or Intranets including e-mail transmissions and information recorded on Websites and the like.

[102] The second part of the application programs 1433 and the corresponding code modules mentioned above may be executed to implement one or more graphical user interfaces (GUIs) to be rendered or otherwise represented upon a display. Through manipulation of typically a keyboard and a mouse, a user of the computer system 1400 and the application may manipulate the interface in a functionally adaptable manner to provide controlling commands and/or input to the applications associated with the GUI(s). Other forms of functionally adaptable user interfaces may also be implemented, such as an audio interface utilizing speech prompts output via loudspeakers and user voice commands input via a microphone.

[103] It is to be understood that the structural context of the computer system 1400 (i.e., the data processing server 140) is presented merely by way of example. Therefore, in some arrangements, one or more features of the computer system 1400 may be omitted. Also, in some arrangements, one or more features of the computer system 1400 may be combined together. Additionally, in some arrangements, one or more features of the computer system 1400 may be split into one or more component parts.

[104] Fig. 9 shows an implementation of the transaction processing server 108 (i.e., the computer system 1300). In this implementation, the transaction processing 108 may be generally described as a physical device comprising at least one processor 802 and at least one memory 804 including computer program codes. The at least one memory 804

and the computer program codes are configured to, with the at least one processor 802, cause the transaction processing server 108 to facilitate the operations described in method 700. The transaction processing server 108 may also include a transaction processing module 806. The memory 804 stores computer program code that the processor 802 compiles to have transaction processing module 806 perform the respective functions.

[105] With reference to Fig. 1, the transaction processing module 806 performs the function of communicating with the requestor device 102 and the provider device 104; and the acquirer server 106 and the issuer server 110 to respectively receive and transmit a transaction, travel request message, a request for data, or other similar messages. Further, the transaction processing module 806 may provide data and information relating to an approved or rejected transaction (e.g. date, time, amount, currency, user name, and other similar data relating to the concerned transaction) to the data processing server 140 and processed into aggregated data. In a case where processing data in relation to a request for data is a service provided by the platform, the transaction processing module 806 may communicate with the data processing server 140 to facilitate payment for the data processing service after data relating to a request for data is retrieved and provided to the requestor.

[106] Fig. 10 shows an alternative implementation of the data processing server 140 (i.e., the computer system 1400). In the alternative implementation, data processing server 140 may be generally described as a physical device comprising at least one processor 902 and at least one memory 904 including computer program codes. The at least one memory 904 and the computer program codes are configured to, with the at least one processor 902, cause the data processing server 140 to perform the operations described in the method 700. The data processing server 140 may also include a data module 906, a comparison module 908, a determination module 910, a calculation module 912 and an aggregation module 914. The memory 904 stores computer program code that the processor 902 compiles to have each of the modules 906 to 914 performs their respective functions.

[107] With reference to Figs. 1 to 7, the comparison module 908 performs the function of comparing a first time range with a second time range, the first time range being based on a start time and an end time indicated in a request for data associated with the start

time and the end time, and the second time range being associated with a first database (e.g. the reference database 150A) or a second database (e.g. the reference database 150B). The reference databases 150A and 150B (including any additional reference databases as explained above) are collectively referred to herein as reference databases 150.

[108] With reference to Figs. 1 to 7, the determination module 910 performs the function of determining one or more sub time ranges for retrieving the data, the one or more sub time ranges being based on the first time range and the comparison by the comparison module 908. The determination module 910 is further configured for determining whether to retrieve data associated with each of the one or more sub time ranges from the first database 150A or the second database 150B based on the comparison.

[109] With reference to Figs. 1 to 7, the calculation module 912 performs the function of for subtracting or adding data retrieved from the first and second databases. Further, the aggregation module 914 performs the function of processing the raw data retrieved from the second database 150B into one or more datapoints comprising information that is aggregated for the concerned time range.

[110] With reference to Figs. 1 to 7, the data module 906 performs the functions of receiving data and information from the requestor device 102, provider device 104, transaction processing server 108, reference databases 150, a cloud and other sources of information to facilitate the method 700. For example, the data module 906 may be configured to receive data and information required for processing the request for data from the requestor device 102, the provider device 104, transaction processing server 108, reference databases 150, and/or other sources of information. The data module 906 may also be configured to retrieve data required by the request from the reference databases 150 including reference databases 150A and 150B, other databases that may be designated by the request for data, and/or other sources of information. The data module 906 may be further configured to send information relating to data retrieved in response to the request for data to the requestor device 102, the provider device 104, the transaction processing server 108, or other destinations where the information is required. The data module 906 may be further configured to communicate with and store data and information for each of the comparison module 908, determination module 910, calculation module 912 and aggregation module 914. Alternatively, all the tasks and functions required for

facilitating the method 700 may be performed by a single processor 902 of the data processing server 140.

[111] Fig. 8B depicts a general-purpose computer system 1500, upon which a combined transaction processing server 108 and data processing server 140 described can be practiced. The computer system 1500 includes a computer module 1501. An external Modulator-Demodulator (Modem) transceiver device 1516 may be used by the computer module 1501 for communicating to and from a communications network 1520 via a connection 1521. The communications network 1520 may be a wide-area network (WAN), such as the Internet, a cellular telecommunications network, or a private WAN. Where the connection 1521 is a telephone line, the modem 1516 may be a traditional "dial-up" modem. Alternatively, where the connection 1521 is a high capacity (e.g., cable) connection, the modem 1516 may be a broadband modem. A wireless modem may also be used for wireless connection to the communications network 1520.

[112] The computer module 1501 typically includes at least one processor unit 1505, and a memory unit 1506. For example, the memory unit 1506 may have semiconductor random access memory (RAM) and semiconductor read only memory (ROM). The computer module 1501 also includes an interface 1508 for the external modem 1516. In some implementations, the modem 1516 may be incorporated within the computer module 1501, for example within the interface 1508. The computer module 1501 also has a local network interface 1511, which permits coupling of the computer system 1500 via a connection 1523 to a local-area communications network 1522, known as a Local Area Network (LAN). As illustrated in Fig. 8D, the local communications network 1522 may also couple to the wide network 1520 via a connection 1524, which would typically include a so-called "firewall" device or device of similar functionality. The local network interface 1511 may comprise an Ethernet circuit card, a Bluetooth® wireless arrangement or an IEEE 802.11 wireless arrangement; however, numerous other types of interfaces may be practiced for the interface 1511.

[113] The I/O interfaces 1508 may afford either or both of serial and parallel connectivity, the former typically being implemented according to the Universal Serial Bus (USB) standards and having corresponding USB connectors (not illustrated). Storage devices 1509 are provided and typically include a hard disk drive (HDD) 1510. Other storage devices such as a floppy disk drive and a magnetic tape drive (not illustrated) may

also be used. An optical disk drive 1512 is typically provided to act as a non-volatile source of data. Portable memory devices, such optical disks, USB-RAM, portable, external hard drives, and floppy disks, for example, may be used as appropriate sources of data to the system 1500.

[114] The components 1505 to 1512 of the computer module 1501 typically communicate via an interconnected bus 1504 and in a manner that results in a conventional mode of operation of the computer system 1500 known to those in the relevant art. For example, the processor 1505 is coupled to the system bus 1504 using a connection 1518. Likewise, the memory 1506 and optical disk drive 1512 are coupled to the system bus 1504 by connections 1519. Examples of computers on which the described arrangements can be practised include IBM-PC's and compatibles, Sun Sparcstations, Apple or like computer systems.

[115] The steps of the method 700 performed by the data processing server 140 and facilitated by the transaction processing server 108 may be implemented using the computer system 1500. For example, the steps of the method 700 as performed by the data processing server 140 may be implemented as one or more software application programs 1533 executable within the computer system 1500. In particular, the steps of the method 700 are effected by instructions in the software 1533 that are carried out within the computer system 1500. The software instructions may be formed as one or more code modules, each for performing one or more particular tasks. The software may also be divided into two separate parts, in which a first part and the corresponding code modules performs the steps of the method 700 and a second part and the corresponding code modules manage a user interface between the first part and the user.

[116] The software may be stored in a computer readable medium, including the storage devices described below, for example. The software is loaded into the computer system 1500 from the computer readable medium, and then executed by the computer system 1500. A computer readable medium having such software or computer program recorded on the computer readable medium is a computer program product. The use of the computer program product in the computer system 1500 preferably effects an advantageous apparatus for a combined transaction processing and data processing server.

[117] The software 1533 is typically stored in the HDD 1510 or the memory 1506. The software is loaded into the computer system 1500 from a computer readable medium, and executed by the computer system 1500. Thus, for example, the software 1533 may be stored on an optically readable disk storage medium (e.g., CD-ROM) 1525 that is read by the optical disk drive 1512. A computer readable medium having such software or computer program recorded on it is a computer program product. The use of the computer program product in the computer system 1500 preferably effects an apparatus for a combined transaction processing and data processing server.

[118] In some instances, the application programs 1533 may be supplied to the user encoded on one or more CD-ROMs 1525 and read via the corresponding drive 1512, or alternatively may be read by the user from the networks 1520 or 1522. Still further, the software can also be loaded into the computer system 1500 from other computer readable media. Computer readable storage media refers to any non-transitory tangible storage medium that provides recorded instructions and/or data to the computer system 1500 for execution and/or processing. Examples of such storage media include floppy disks, magnetic tape, optical disc, a hard disk drive, a ROM or integrated circuit, USB memory, a magneto-optical disk, or a computer readable card such as a PCMCIA card and the like, whether or not such devices are internal or external of the computer module 1501. Examples of transitory or non-tangible computer readable transmission media that may also participate in the provision of software, application programs, instructions and/or data to the computer module 1501 include radio or infra-red transmission channels as well as a network connection to another computer or networked device, and the Internet or Intranets including e-mail transmissions and information recorded on Websites and the like.

[119] The second part of the application programs 1533 and the corresponding code modules mentioned above may be executed to implement one or more graphical user interfaces (GUIs) to be rendered or otherwise represented upon a display. Through manipulation of typically a keyboard and a mouse, a user of the computer system 1500 and the application may manipulate the interface in a functionally adaptable manner to provide controlling commands and/or input to the applications associated with the GUI(s). Other forms of functionally adaptable user interfaces may also be implemented, such as an audio interface utilizing speech prompts output via loudspeakers and user voice commands input via a microphone.

[120] It is to be understood that the structural context of the computer system 1500 (i.e., combined transaction processing and data processing server 1500) is presented merely by way of example. Therefore, in some arrangements, one or more features of the server 1500 may be omitted. Also, in some arrangements, one or more features of the server 1500 may be combined together. Additionally, in some arrangements, one or more features of the server 1500 may be split into one or more component parts.

[121] Fig. 11 shows an alternative implementation of combined transaction processing and data processing server (i.e., the computer system 1500). In the alternative implementation, the combined transaction processing and data processing server may be generally described as a physical device comprising at least one processor 1002 and at least one memory 904 including computer program codes. The at least one memory 1004 and the computer program codes are configured to, with the at least one processor 1002, cause the combined transaction processing and data processing server to perform the operations described in the steps of the method 700. The combined transaction processing and data processing server may also include a transaction processing module 806, a data module 906, a comparison module 908, a determination module 910, a calculation module 912 and an aggregation module 914. The memory 1004 stores computer program code that the processor 1002 compiles to have each of the modules 806 to 912 performs their respective functions. The transaction processing module 806 performs the same functions as described for the same transaction processing module in Fig. 9. The data module 906, comparison module 908, determination module 910, calculation module 912 and aggregation module 914 perform the same functions as described for the same corresponding modules in Fig. 10.

[122] It will be appreciated by a person skilled in the art that numerous variations and/or modifications may be made to the present disclosure as shown in the specific embodiments without departing from the scope of the specification as broadly described. The present embodiments are, therefore, to be considered in all respects to be illustrative and not restrictive.

CLAIMS

What is claimed is:

1. A method for adaptively processing a request for data, comprising:

comparing, by a processor, a first time range with a second time range, the first time range being based on a start time and an end time indicated in a request for data associated with the start time and the end time, and the second time range being associated with a first database or a second database, the first database storing aggregated data and the second database storing raw data;

determining, by the processor, one or more sub time ranges for retrieving the data, the one or more sub time ranges being based on the first time range and the comparison;

assessing, by the processor, whether to retrieve data associated with each of the one or more sub time ranges from the first database or the second database based on the comparison; and

retrieving the data based on the assessment.

2. The method of claim 1, wherein the second database stores data in one or more segments, each segment comprising data within a predetermined time range, wherein the first time range starts from the start time to a current time at which the comparison is made and the second time range is the predetermined time range of the second database,

wherein determining the one or more sub time ranges comprises determining a sub time range that starts from the start time to the end time if the first time range is shorter than the second time range, and

wherein assessing whether to retrieve data associated with each of the one or more sub time ranges from the first database or the second database further comprises determining that the data associated with the sub time range is to be retrieved from the second database.

3. The method of claim 1, wherein the second database stores data in one or more segments, each segment comprising data within a predetermined time range, wherein the first time range starts from the start time to a current time at which the comparison is made and is shorter than the predetermined time range of the second database, and the second time range is an amount of time in which data is kept in the first database before being deleted from the first database, wherein determining the one or more sub time ranges

further comprises determining whether the first time range is shorter than the second time range.

4. The method of claim 1, wherein the first time range starts from the start time to a current time at which the comparison is made and the second time range is a first amount of time in which data is kept in the first database before being deleted from the first database, wherein comparing the first time range with the second time range further comprises:

comparing the first time range with a third time range if the first time range is longer than the second time range, wherein the third time range is a second amount of time in which data is kept in the first database before being deleted from the first database, the third time range being longer than the second time range; and

determining the one or more sub time ranges based on the comparison.

5. The method of claim 1, wherein determining the one or more sub time ranges further comprises determining a first sub time range and a second sub time range, the first time range being an addition of the first sub time range with the second sub time range, the method further comprising:

retrieving data associated with each of the first and second sub time range from the first database or second database based on the comparison; and

adding the retrieved data together.

6. The method of claim 1, wherein determining the one or more sub time ranges further comprises determining a first sub time range and a second sub time range, the first time range being a subtraction of the second sub time range from the first sub time range, the method further comprising:

retrieving data associated with the first sub time range from the first database, and retrieving data associated with the second sub time range from the second database based on the comparison; and

subtracting the data retrieved from the second database from the data retrieved from the first database.

7. The method of claim 6, wherein the first database stores aggregated data comprising a plurality of datapoints, each datapoint comprising information that is aggregated for a predetermined time range, wherein

retrieving data associated with the first sub time range from the first database comprises retrieving a plurality of datapoints associated with the first sub time range from the first database, and retrieving data associated with the second sub time range from the second database comprises retrieving raw data associated with the second sub time range from the second database; wherein

subtracting the data retrieved from the second database from the data retrieved from the first database further comprises:

processing the raw data retrieved from the second database into one or more datapoints comprising information that is aggregated for the second sub time range; and

subtracting the one or more datapoints processed from raw data retrieved from the second database from the plurality of datapoints associated with the first sub time range.

8. A system for adaptively processing a request for data, comprising:

at least one processor; and

at least one memory including computer program code;

the at least one memory and the computer program code configured to, with the at least one processor, cause the system at least to:

compare, by a processor, a first time range with a second time range, the first time range being based on a start time and an end time indicated in a request for data associated with the start time and the end time, and the second time range being associated with a first database or a second database, the first database storing aggregated data and the second database storing raw data;

determine, by the processor, one or more sub time ranges for retrieving the data, the one or more sub time ranges being based on the first time range and the comparison;

assess, by the processor, whether to retrieve data associated with each of the one or more sub time ranges from the first database or the second database based on the comparison; and

retrieve the data based on the assessment.

9. The system of claim 8, wherein the second database stores data in one or more segments, each segment comprising data within a predetermined time range, wherein the first time range starts from the start time to a current time at which the comparison is made

and the second time range is the predetermined time range of the second database, wherein

determining the one or more sub time ranges comprises determining a sub time range that starts from the start time to the end time if the first time range is shorter than the second time range, and wherein

assessing whether to retrieve data associated with each of the one or more sub time ranges from the first database or the second database further comprises determining that the data associated with the sub time range is to be retrieved from the second database.

10. The system of claim 8, wherein the second database stores data in one or more segments, each segment comprising data within a predetermined time range, wherein the first time range starts from the start time to a current time at which the comparison is made and is shorter than the predetermined time range of the second database, and the second time range is an amount of time in which data is kept in the first database before being deleted from the first database, wherein determining the one or more sub time ranges further comprises determining whether the first-time range is shorter than the second time range.

11. The system of claim 8, wherein the first time range starts from the start time to a current time at which the comparison is made and the second time range is a first amount of time in which data is kept in the first database before being deleted from the first database, wherein comparing the first time range with the second time range further comprises:

comparing the first time range with a third time range if the first time range is longer than the second time range, wherein the third time range is a second amount of time in which data is kept in the first database before being deleted from the first database, the third time range being longer than the second time range; and

determining the one or more sub time ranges based on the comparison.

12. The system of claim 8, wherein determining the one or more sub time ranges further comprises determining a first sub time range and a second sub time range, the first time range being an addition of the first sub time range with the second sub time range, the system further configured to:

retrieve data associated with each of the first and second sub time range from the first database or second database based on the comparison; and

add the retrieved data together.

13. The system of claim 8, wherein determining the one or more sub time ranges further comprises determining a first sub time range and a second sub time range, the first time range being a subtraction of the second sub time range from the first sub time range, the system further configured to:

retrieve data associated with the first sub time range from the first database, and retrieve data associated with the second sub time range from the second database based on the comparison; and

subtract the data retrieved from the second database from the data retrieved from the first database.

14. The system of claim 13, wherein the first database stores data comprising a plurality of datapoints, each datapoint comprising information that is aggregated for an associated time range, and the second database stores raw data, wherein

retrieving data associated with the first sub time range from the first database comprises retrieving a plurality of datapoints associated with the first sub time range from the first database, and retrieving data associated with the second sub time range from the second database comprises retrieving raw data associated with the second sub time range from the second database; wherein

subtracting the data retrieved from the second database from the data retrieved from the first database further comprises:

processing the raw data retrieved from the second database into one or more datapoints comprising information that is aggregated for the second sub time range; and

subtracting the one or more datapoints processed from raw data retrieved from the second database from the plurality of datapoints associated with the first sub time range.