



(51) International Patent Classification:
G06F 17/30 (2006.01)

(21) International Application Number:
PCT/US2014/044932

(22) International Filing Date:
30 June 2014 (30.06.2014)

(25) Filing Language: English

(26) Publication Language: English

(71) Applicant: **HEWLETT-PACKARD DEVELOPMENT COMPANY, LP** [US/US]; 11445 Compaq Center Drive W, Houston, Texas 77070 (US).

(72) Inventors: **SIMITSIS, Alkiviadis**; 1501 Page Mill Rd., Palo Alto, California 94304-1100 (US). **WILKINSON, William, K.**; 1501 Page Mill Rd., Palo Alto, California 94304-1100 (US).

(74) Agents: **ESTEBAN, Michelle** et al.; Hewlett-Packard Company, Intellectual Property Administration, 3404 E. Harmony Road, Mail Stop 35, Fort Collins, California 80528-9599 (US).

(81) Designated States (*unless otherwise indicated, for every kind of national protection available*): AE, AG, AL, AM, AO, AT, AU, AZ, BA, BB, BG, BH, BN, BR, BW, BY,

BZ, CA, CH, CL, CN, CO, CR, CU, CZ, DE, DK, DM, DO, DZ, EC, EE, EG, ES, FI, GB, GD, GE, GH, GM, GT, HN, HR, HU, ID, IL, IN, IR, IS, JP, KE, KG, KN, KP, KR, KZ, LA, LC, LK, LR, LS, LT, LU, LY, MA, MD, ME, MG, MK, MN, MW, MX, MY, MZ, NA, NG, NI, NO, NZ, OM, PA, PE, PG, PH, PL, PT, QA, RO, RS, RU, RW, SA, SC, SD, SE, SG, SK, SL, SM, ST, SV, SY, TH, TJ, TM, TN, TR, TT, TZ, UA, UG, US, UZ, VC, VN, ZA, ZM, ZW.

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

Declarations under Rule 4.17:

— *as to the identity of the inventor (Rule 4.17(i))*

Published:

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

(54) Title: AUTOMATIC GENERATION OF SUB-QUERIES

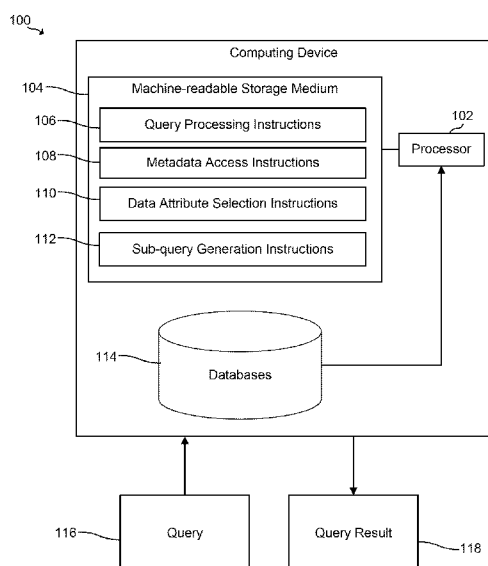


FIG. 1

(57) Abstract: Example implementations relate to automatic generation of sub-queries. For example, a computing device may include a processor. The processor may receive a query defining a particular set of data stored in at least one database. The processor may access metadata associated with the particular set of data and may select a data attribute associated with the particular set of data based on the metadata. The processor may automatically generate sub-queries of the query using the data attribute selected based on the metadata. The processor may process the sub-queries and provide a query result of the query that is a combination of sub-query results of the sub-queries.

AUTOMATIC GENERATION OF SUB-QUERIES

BACKGROUND

[0001] Many entities utilize databases for storage of data relating to the entity. For example, a company may store data relating to the company's sales. Data stored in these databases may be accessed and analyzed for various purposes.

BRIEF DESCRIPTION OF THE DRAWINGS

[0002] Some examples of the present application are described with respect to the following figures:

[0003] FIG. 1 is a block diagram of an example computing device for automatically generating sub-queries;

[0004] FIG. 2 is a flowchart illustrating an example method of automatically generating sub-queries;

[0005] FIG. 3 is a diagram of example database tables of a database that may be used to automatically generate sub-queries; and

[0006] FIG. 4 is a diagram of an example histogram of data that may be used to automatically generate sub-queries.

DETAILED DESCRIPTION

[0007] As described above, data stored in a database may be accessed and analyzed for various purposes. A query of the data may indicate a particular set of data that is being requested. For example, an employee of a grocery store may wish to analyze sales for a particular month and may query the database for this particular set of sales data. However, for databases containing large amounts of data, a query for a set of data may be time-consuming and may utilize a large amount of memory resources while the query is being run.

[0008] To more efficiently run a query of data stored in one or more databases, the query may be automatically transformed into a functionally-equivalent form using a workflow of smaller sub-queries. Each sub-query

may process a disjoint subset of the input to the original query, each providing a partial result of the original query. The results of each sub-query may be combined to produce a result identical to the result that would have been produced by running the original query itself. The sub-queries may each maintain the same general structure as the original query but may be collectively performed faster than performing the original query.

[0009] The automatic generation of sub-queries may be performed in response to receiving a query for a particular set of data stored in any suitable database. For example, the automatic generation of sub-queries may be performed for long-running, analytic queries (e.g., in a declarative query language) over a relational (e.g., SQL) database. While the automatic generation of sub-queries is described in the context of example databases, the technology may be extended to queries or flows spanning multiple engines as well, such as dataflow engines, workflow engines, time to live (TTL) flow engines, massively parallel processing engines, and the like. For example, a flow spanning a Hadoop engine and a database engine may use data partitioning either for the entire flow (e.g., using a merge operation toward the end of the flow) or across engines.

[0010] The following is an example of using automatically-generated sub-queries to process a query for a particular set of data stored in a database. For example, assume a database that stores fact tables and dimension tables. Fact tables may be relatively larger than dimension tables and may frequently receive data describing real-world events. For example, a fact table may include data indicating that customer X purchased item Y at store Z at time W in quantity Q for price P. A fact table may record measures of an event (e.g., quantity, price, etc.) while other descriptive attributes of the event (e.g., customer name, age, store location, day and time, item color, size, weight, etc.) may be recorded in dimension tables that may be referenced by the fact table. The fact table may contain key values that reference the descriptive attributes recorded in dimension tables. In some examples, a key value in a fact table may be an integer that may be mapped to a corresponding entry in a dimension table. A dimension table may be relatively stable compared to a fact table and may be slow-changing (e.g., store

information), while a fact table may change more quickly as new fact events are received more frequently (e.g., a customer purchase). In some examples, dimension tables may be smaller than fact tables. This type of database organization may sometimes be referred to as a star schema because a central fact table may be linked (e.g., by key values) to dimension tables.

[0011] In this example, the dimension tables may include a customer table containing customer attribute information (e.g., customer name, address, account number, etc.), an item table containing item attribute information (e.g., item color, size, price, etc.), a store table containing store attribute information (e.g., store address, store type, etc.), and a date table containing date attribute information (e.g., time of day, day, month, year, etc.). A star query may be performed to retrieve all events with certain specified attributes. When the query is received, the query is altered in a parameterized manner such that each value of the parameter creates a query variant that returns a disjoint subset of the original query result and such that these intermediate results may be combined to form the complete result of the original query. In some examples, the key values that link the fact table to the dimension tables may be used to automatically generate sub-queries. For example, assume that the key values are integer values. A dimension table used in the query may be selected, and a filter condition may be added to the key value related to the selected dimension table. For example, suppose the original star query is:

Select price, quantity from SalesFact where dateKey>d1 and dateKey<d2

[0012] This may be rewritten into sub-queries as:

*Select price, quantity from SalesFact where dateKey>d1 and dateKey <d2
and (dateKey modulo 10) = 0*

[0013] Assuming roughly equal distribution of dates, the sub-queries may return approximately ten percent of the results of the original query. The remaining query results may be obtained by generating nine additional queries using parameter values 1, 2, ..., 9 for the matching value of the

modulo function. Assuming these queries store their immediate results in temporary tables in the database, the final query result may be obtained by concatenating the temporary tables. Any sort order, limit clause, and the like that is to be implemented on the original query results may be replicated on the final query result from the sub-queries.

[0014] In some examples, if the original query contains filter conditions on other attributes (e.g., store region = California), the sub-query approach may still be performed. However, in this case, the query technology may choose which attribute to parameterize (e.g., an attribute from a dimension table, a key value attribute, etc.). The attribute to be parameterized may be selected using metadata associated with the table for the attribute. The metadata may indicate properties of the data in a table (e.g., cardinality of referenced attributes, uniqueness, etc.), and these properties may be used to determine which attribute to select for parameterization. For example, the metadata may indicate that generating sub-queries based on parameterizing store locations is more efficient than parameterizing other attributes, and as such, the sub-queries may be generated based on store locations.

[0015] While the example above was described in terms of fact and dimension tables, one of ordinary skill in the art will recognize that a similar approach may be used for any arbitrary analytic queries and table organizations by analyzing the query structure and using table metadata associated with the database used. In some examples, the query structure may be obtained from the database system explain plan, and the query metadata may be obtained from the database system.

[0016] Referring now to the figures, FIG. 1 is a block diagram of an example computing device 100 for automatically generating sub-queries. The sub-queries may be automatically generated in response to receiving a query requesting a particular set of data stored in databases 114.

[0017] Computing device 100 may be, for example, a web-based server, a local area network server, a cloud-based server, a notebook computer, a desktop computer, an all-in-one system, a tablet computing device, a mobile phone, an electronic book reader, or any other electronic device suitable for automatically generating sub-queries. Computing device 100 may include a

processor 102 and a machine-readable storage medium 104. Computing device 100 may be in communication with one or more additional computing devices. Computing device 100 may receive a query 116 (e.g., from a user operating computing device 100, from the one or more additional computing devices in communication with the computing device 100, etc.) and may return a query result 118 (e.g., to the user operating computing device 100, to the one or more additional computing devices in communication with the computing device 100, etc.).

[0018] Processor 102 may be a central processing unit (CPU), a semiconductor-based microprocessor, and/or other hardware devices suitable for retrieval and execution of instructions stored in machine-readable storage medium 104. Processor 102 may fetch, decode, and execute instructions 106, 108, 110, and 112 to control a process of automatically generating sub-queries. As an alternative or in addition to retrieving and executing instructions, processor 102 may include at least one electronic circuit that includes electronic components for performing the functionality of instructions 106, 108, 110, 112, or a combination thereof.

[0019] Machine-readable storage medium 104 may be any electronic, magnetic, optical, or other physical storage device that contains or stores executable instructions. Thus, machine-readable storage medium 104 may be, for example, Random Access Memory (RAM), an Electrically Erasable Programmable Read-Only Memory (EEPROM), a storage device, an optical disc, and the like. In some examples, machine-readable storage medium 104 may be a non-transitory storage medium, where the term “non-transitory” does not encompass transitory propagating signals. As described in detail below, machine-readable storage medium 104 may be encoded with a series of processor executable instructions 106, 108, 110, and 112 for receiving a query 116 defining a particular set of data stored in the databases 114, accessing metadata associated with the particular set of data, selecting a data attribute associated with the particular set of data based on the metadata, automatically generating sub-queries of the query 116 using the data attribute selected, processing the sub-queries, and providing a query result 118 of the query 116, where the query result 118 is a combination of sub-query results of the sub-query.

[0020] Query processing instructions 106 may receive a query defining a particular set of data stored in databases 114. For example, query processing instructions 106 may receive a query requesting all sales data from May of the current year from all stores in California.

[0021] Metadata access instructions 108 may access metadata associated with the particular set of data defined by the query that is received by query processing instructions 106. The metadata may include any metadata associated with data in databases 114. For example, metadata access instructions 108 may access metadata relating to data tables that may be accessed for a query requesting all sales data from May of the current year from all stores in California, such as the number of unique items in the data tables, the number of attributes in the data tables, and the like.

[0022] Data attribute selection instructions 110 may select, identify, or determine, based on the metadata accessed by metadata access instructions 108, a data attribute associated with the particular set of data. A data attribute may be any characteristic, category, classification, or type of information that may be used to identify data. The selection of the data attribute may be based on metadata indicating that parameterization of that particular data attribute may result in sub-queries that may be processed relatively efficiently. For example, the metadata may indicate that parameterizing the date attribute (e.g., date, store, etc.) may result in efficient sub-queries, and as such, data attribute selection instructions 110 may select the data attribute for parameterization. In some examples, the data attribute selected may result in sub-queries that each partition a disjoint subset of the original query. In some examples, the data attribute selected is such that each sub-query produces a result that is approximately equal in size to the results of the other sub-queries. In some examples, the data attribute selected is such that the runtime for processing and combining the sub-queries is less than the runtime for processing the original query.

[0023] Sub-query generation instructions 112 may use the selected data attribute to automatically generate or create sub-queries of the query. For example, if the date attribute was selected to be parameterized daily, sub-queries corresponding to disjoint increments or ranges of data associated with

the date attribute may be generated such that a sub-query for each day in the month of May of the current year may be generated.

[0024] Once the sub-queries are generated, query processing instructions 106 may analyze and process the sub-queries, combine and calculate the results of each sub-query, and provide the combined result as the result to the original query.

[0025] Databases 114 may include any suitable number and type of database. For example, databases 114 may include a star schema database containing fact tables and dimension tables.

[0026] FIG. 2 is a flowchart illustrating an example method 200 of automatically generating sub-queries. Method 200 may be implemented using computing device 100 of FIG. 1.

[0027] Method 200 includes, at 202, receiving a query defining a particular set of data stored in at least one database. For example, the query may define a set of data requested.

[0028] Method 200 also includes, at 204, accessing metadata associated with the particular set of data. The metadata may be accessed in any suitable manner. In some examples, the metadata may be stored with the database and may be accessed from the database. In some examples, a subset of the query may be processed to obtain metadata associated with the subset of the query. In some examples, the query may be processed and then stopped before completion in order to obtain relevant metadata. In some examples, a query for a subset of data may be processed to obtain relevant metadata.

[0029] Method 200 also includes, at 206, selecting a data attribute associated with the particular set of data based on the metadata. For example, the metadata may be used to determine which data attribute to use in the creation of sub-queries, to determine how to parameterize the attribute (e.g., based on the attribute data itself, based on key values associated with the attribute data in dimension tables, based on a histogram of the attribute data, etc.), and the like.

[0030] Method 200 also includes, at 208, automatically generating sub-queries of the query using the data attribute selected. The sub-queries may

be automatically generated in any suitable manner. In some examples, the original query may be decomposed by partitioning the query input data based on the selected data attribute.

[0031] Method 200 also includes, at 210, processing the sub-queries. The sub-queries may be processed in any suitable manner. For example, the sub-queries may be processed by directly by an interpreter, indirectly after compilation into native machine language, and the like.

[0032] Method 200 also includes, at 212, providing a query result of the query, the query result being a combination of sub-query results of the sub-query. In some examples, the manner in which the sub-query results are combined may depend on the original query. For example, the sub-query results may be combined such that the combination of the results is the same as the results that the original query would have produced if processed in its entirety. The sub-query results may be combined in any manner (e.g., concatenation, merge operation, merge sort, limiting operation, etc.).

[0033] FIG. 3 is a diagram of example database tables of a database 114 that may be used to automatically generate sub-queries. The database 114 may include fact table 302 and any number of dimension tables, such as store table 304, customer table 306, and date table 308. The database 114 may also include metadata associated with each of the tables in the database 114.

[0034] In the example of FIG. 3, fact table 302 may include information about sales for a company. The sales information in fact table 302 may include data about each sale for the company, including store information, customer information, date information, quantity information, amount information, and the like. For example, for a given sale 314, fact table 302 may include data relating to the sale 314, such as data indicating that the sale 314 was made in Store A by Customer B on Date C and that Quantity D was purchased in the amount of Amount E. The data relating to sale 314 may be included in fact table 302 in any suitable manner, such as including the actual information (e.g., Amount E is \$13.42), a key value referencing data in a related dimension table, and the like.

[0035] Store table 304, customer table 306, and date table 308 may each be a dimensional table containing data associated with a column in fact table

302. For example, store table 304 may include any data associated with a company's stores (e.g., store name, address, location, size, employees, etc.). Data in the quantity and amount columns may be measurements that are included in fact table 302.

[0036] When a query for data in fact table 302 is received, sub-queries may be automatically generated based on the metadata associated with each of the tables in database 114. For example, a query requesting sales information for California stores in May of the current year may be received. The query may be automatically partitioned into thirty-one sub-queries based on the metadata (e.g., a sub-query for each day in May of the current year). These sub-queries may each be processed to return the sales for California stores on May 1st, May 2nd, May 3rd, and the like. The results from each sub-query may be combined, and the combined result may be returned as the result of the originally-received query. Sub-queries may be generated based on the metadata in database 114 using any data attribute for parameterization, such as generating sub-queries based on a date range (e.g., daily), a store location (e.g., by city), customers (e.g., by letter of last name), key values for a dimension table (e.g., key value ranges), a histogram of data in a dimension table (e.g., a histogram of sales in a given day), and the like.

[0037] FIG. 4 is a diagram of an example histogram 400 of data that may be used to automatically generate sub-queries. Histogram 400 shows curve 402 for the number of sales at different times of a particular day. For example, curve 402 shows that the number of sales at the beginning and end of the day are each less than the number of sales that occur in the middle of the day. The metadata associated with sales data may indicate that the distribution of sales is not equal for each hour of the day. As such, generating sub-queries by parameterizing sales by the hour may not be efficient, since processing a sub-query for the 11am hour would take longer than processing a sub-query for the 6am hour. Instead, the histogram 400 may be used for generating sub-queries based on curve 402. For example, a sub-query may be generated based on the area under curve 402 such that the sales are equally distributed between the sub-queries generated. For example, if the area under curve 402 between 6am and 8am is equal to the area under curve

402 between 11am and 12pm, a sub-query may be generated for each of these time ranges such that processing of each of these sub-queries takes approximately the same amount of time.

[0038] Examples provided herein (e.g., methods) may be implemented in hardware, software, or a combination of both. Example systems may include a controller/processor and memory resources for executing instructions stored in a tangible non-transitory medium (e.g., volatile memory, non-volatile memory, and/or machine-readable media). Non-transitory machine-readable media can be tangible and have machine-readable instructions stored thereon that are executable by a processor to implement examples according to the present disclosure.

[0039] An example system can include and/or receive a tangible non-transitory machine-readable medium storing a set of machine-readable instructions (e.g., software). As used herein, the controller/processor can include one or a plurality of processors such as in a parallel processing system. The memory can include memory addressable by the processor for execution of machine-readable instructions. The machine-readable medium can include volatile and/or non-volatile memory such as a random access memory ("RAM"), magnetic memory such as a hard disk, floppy disk, and/or tape memory, a solid state drive ("SSD"), flash memory, phase change memory, and so on.

Claims

What is claimed is:

1. A computing device comprising:
a processor to:
receive a query defining a particular set of data stored in at least one database;
access metadata associated with the particular set of data;
select a data attribute associated with the particular set of data based on the metadata;
automatically generate sub-queries of the query using the data attribute selected based on the metadata;
process the sub-queries; and
provide a query result of the query, the query result being a combination of sub-query results of the sub-queries.
2. The computing device of claim 1, wherein selecting the data attribute based on the metadata includes selecting the data attribute based on at least one of processing time of each sub-query, distribution of data between the sub-queries, a number of results for each sub-query, and user-specified criteria associated with the data attribute, wherein the sub-queries produce a disjoint subset of an original result of the query.
3. The computing device of claim 1, wherein the sub-queries correspond to disjoint ranges of data associated with the data attribute.
4. The computing device of claim 1, wherein the sub-queries correspond to disjoint ranges of key values relating to data associated with the data attribute.
5. The computing device of claim 1, wherein generating the sub-queries is based on a histogram of data associated with the data attribute.

6. A method comprising:
receiving, by a computing device, a query defining a particular set of data stored in at least one database;
accessing, by the computing device, metadata associated with the particular set of data;
identifying, by the computing device, a data attribute associated with the particular set of data based on the metadata;
automatically creating, by the computing device, sub-queries of the query using the data attribute identified based on the metadata;
analyzing, by the computing device, the sub-queries; and
calculating, by the computing device, a query result of the query, the query result including sub-query results of the sub-queries.
7. The method of claim 6, further comprising:
identifying the data attribute based on at least one of processing time of each sub-query, distribution of data between the sub-queries, a number of results for each sub-query, and user-specified criteria associated with the data attribute, wherein the sub-queries produce a disjoint subset of an original result of the query.
8. The method of claim 6, wherein the sub-queries correspond to disjoint ranges of data associated with the data attribute.
9. The method of claim 6, wherein the sub-queries correspond to disjoint ranges of key values relating to data associated with the data attribute.
10. The method of claim 6, wherein generating the sub-queries is based on a histogram of data associated with the data attribute.

11. A non-transitory machine-readable storage medium storing instructions that, if executed by at least one processor of a computing device, cause the computing device to:

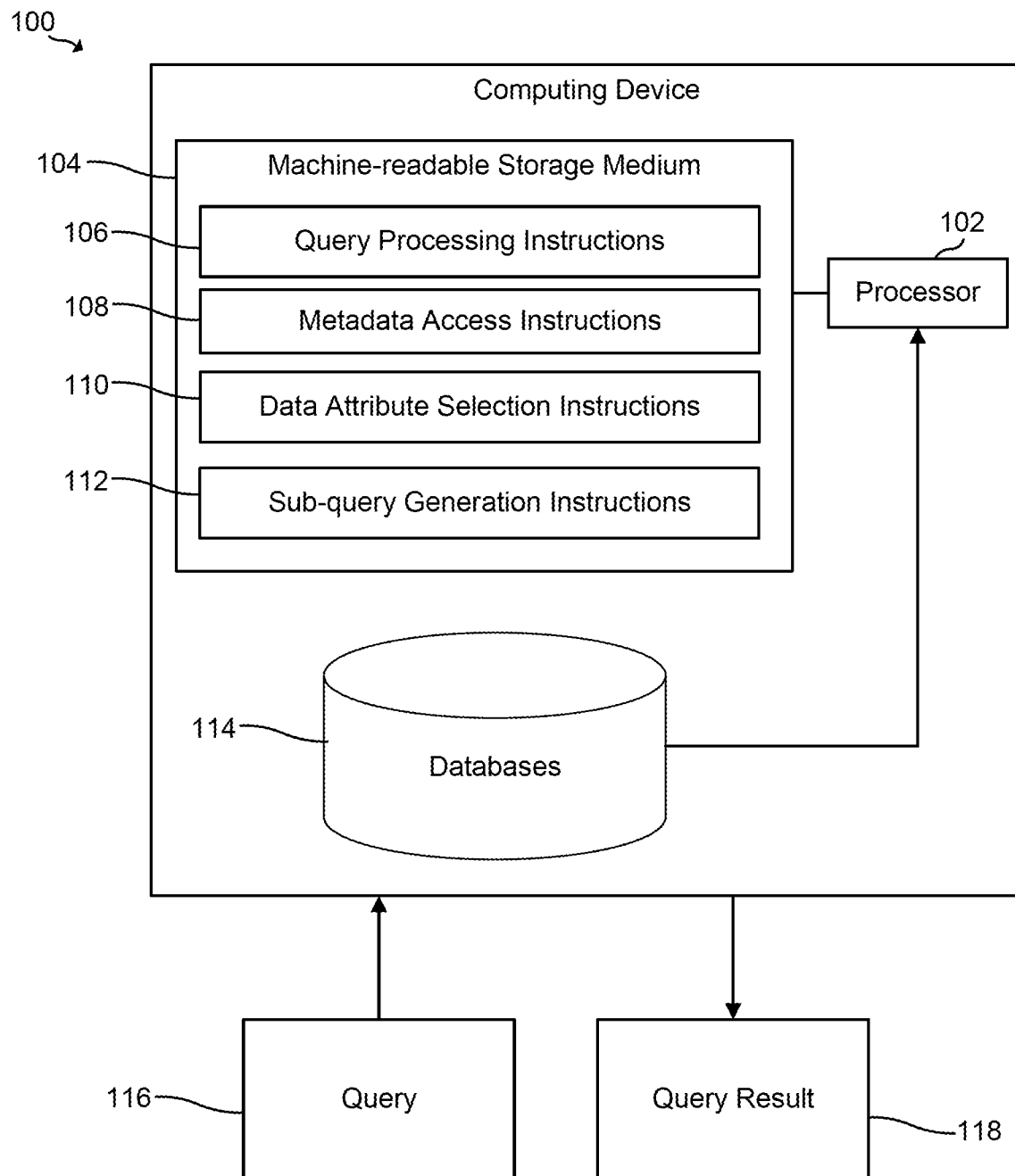
- receive a query defining a particular set of data stored in at least one database;
- access metadata associated with the particular set of data;
- determine a data attribute associated with the particular set of data based on the metadata;
- automatically generate sub-queries of the query using the data attribute determined based on the metadata;
- perform the sub-queries; and
- determine a query result of the query, the query result being a combination of sub-query results of the sub-queries.

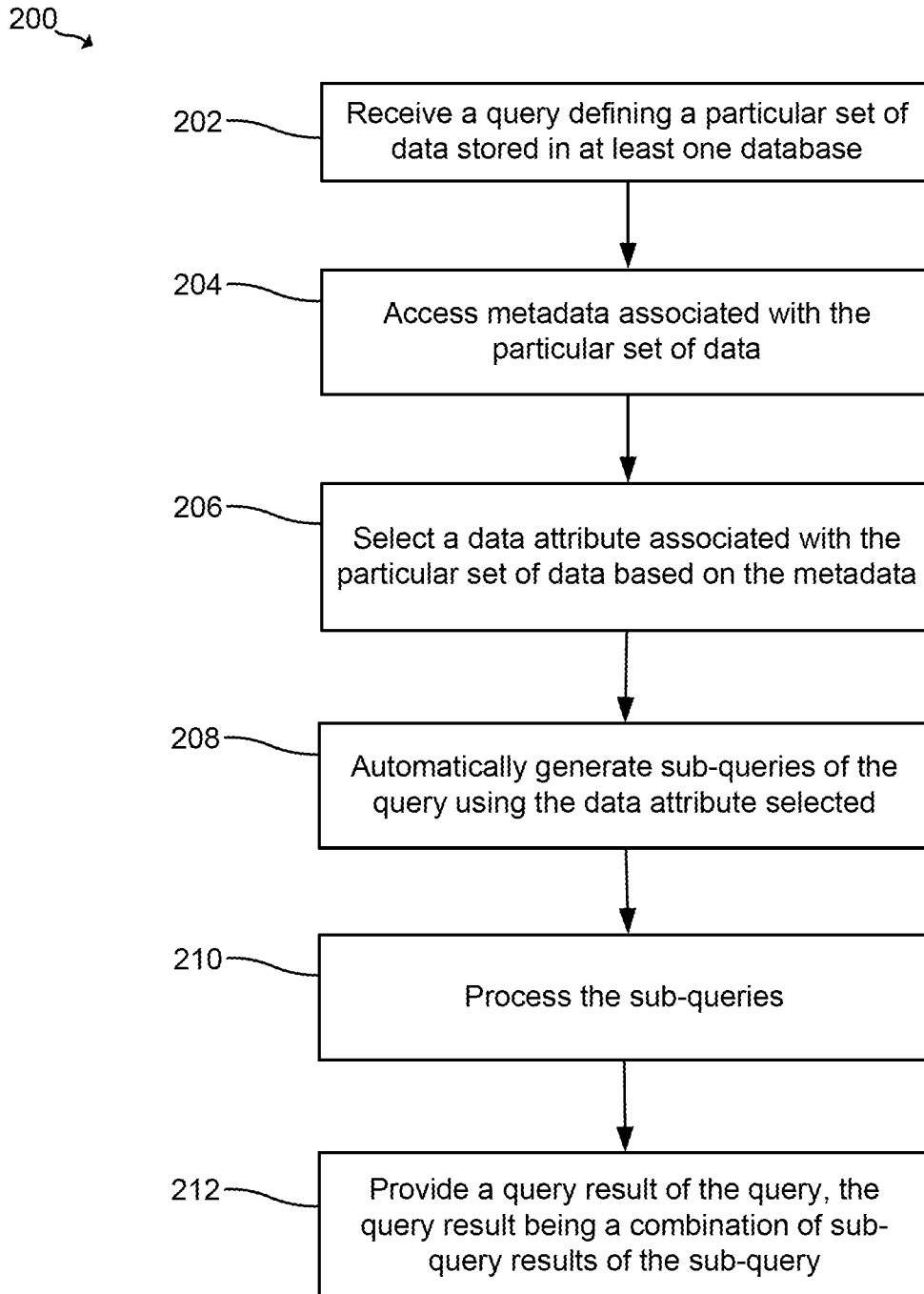
12. The non-transitory machine-readable storage medium of claim 11, wherein the instructions, if executed by the at least one processor, further cause the computing device to determine the data attribute based on at least one of processing time of each sub-query, distribution of data between the sub-queries, a number of results for each sub-query, and user-specified criteria associated with the data attribute, wherein the sub-queries produce a disjoint subset of an original result of the query.

13. The non-transitory machine-readable storage medium of claim 11, wherein the sub-queries correspond to disjoint ranges of data associated with the data attribute.

14. The non-transitory machine-readable storage medium of claim 11, wherein the sub-queries correspond to disjoint ranges of key values relating to data associated with the data attribute.

15. The non-transitory machine-readable storage medium of claim 11, wherein generating the sub-queries is based on a histogram of data associated with the data attribute.

**FIG. 1**

**FIG. 2**

114 →

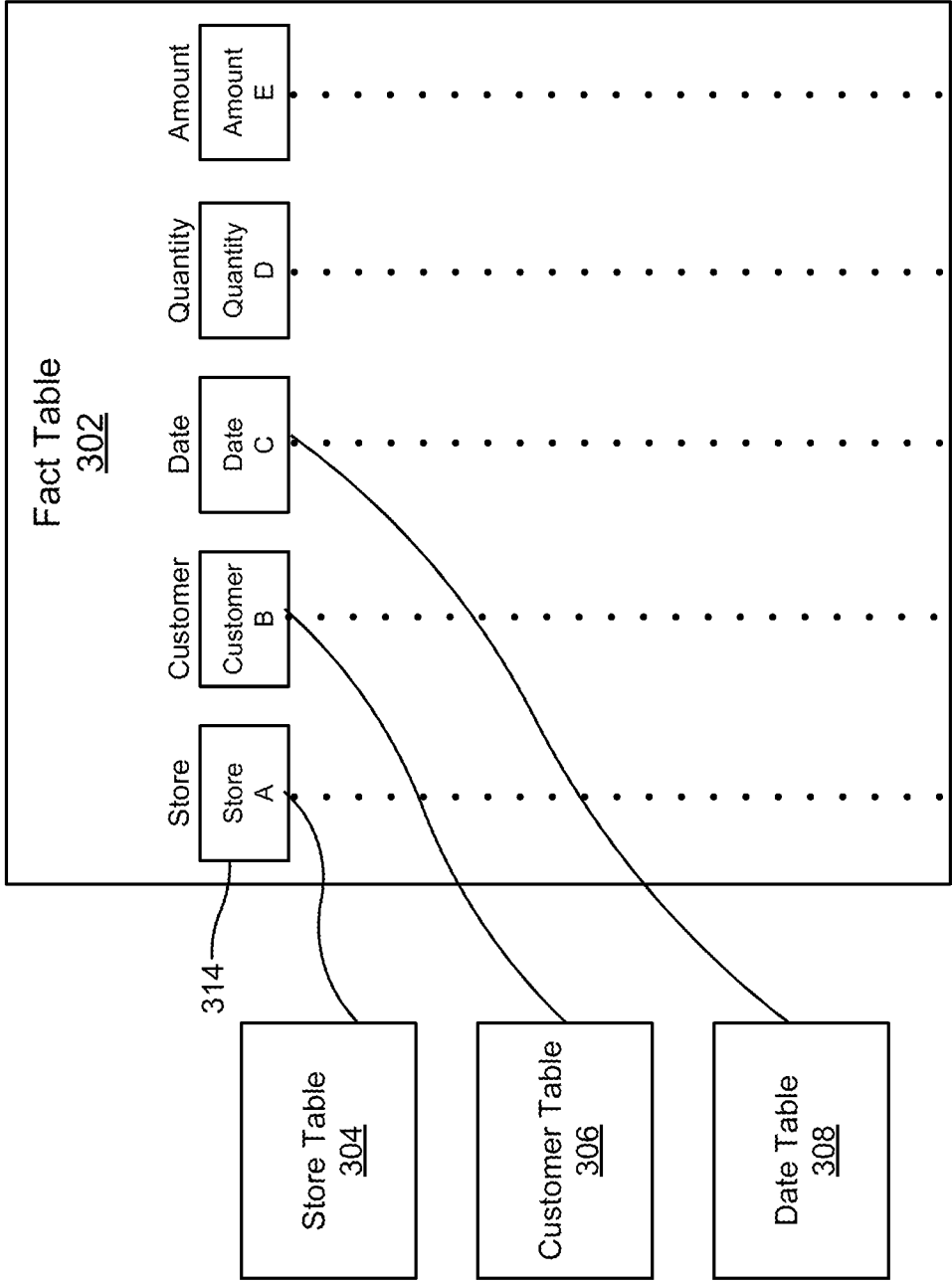


FIG. 3

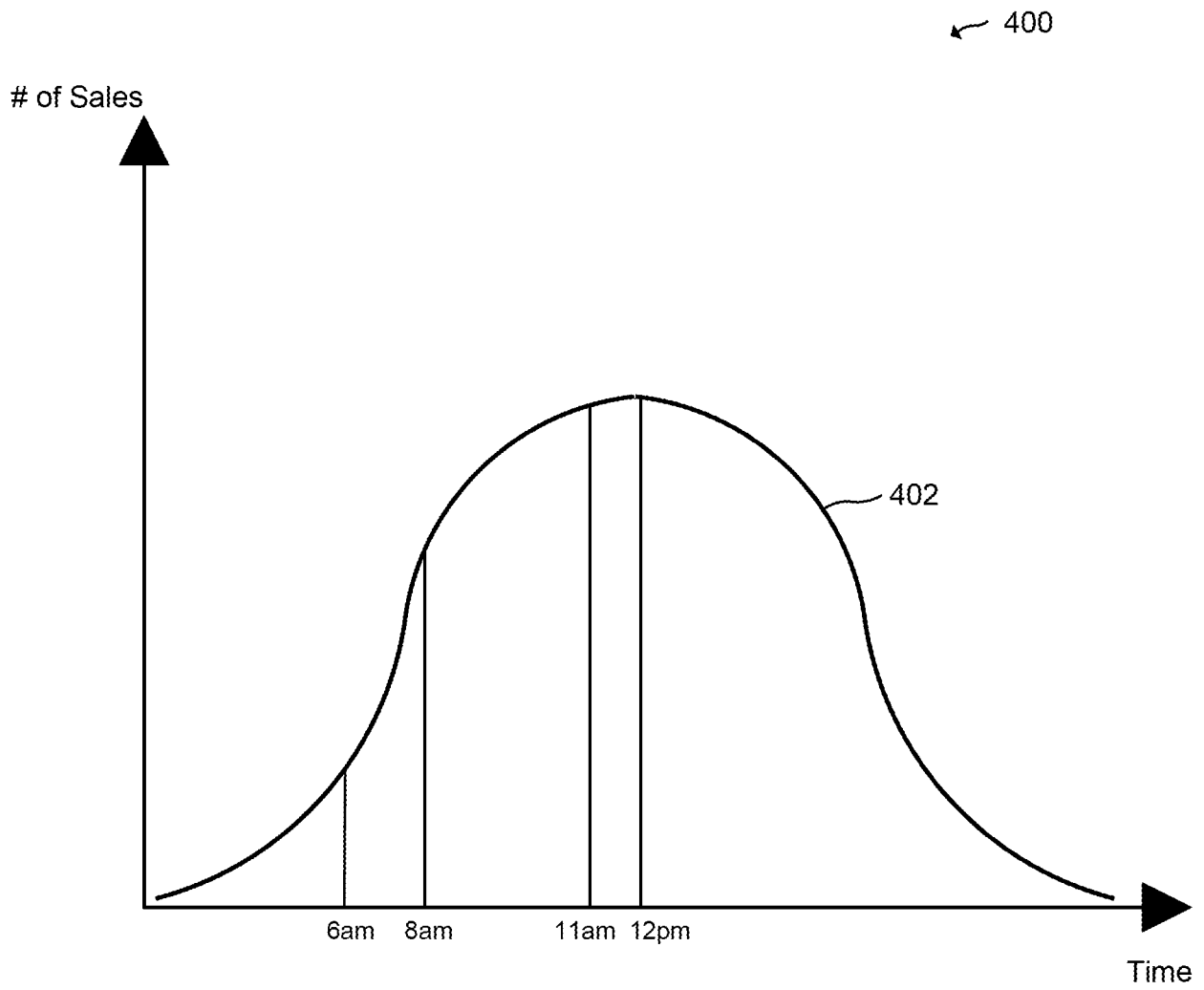


FIG. 4

INTERNATIONAL SEARCH REPORT

International application No.
PCT/US2014/044932**A. CLASSIFICATION OF SUBJECT MATTER****G06F 17/30(2006.01)i**

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

B. FIELDS SEARCHED

Minimum documentation searched (classification system followed by classification symbols)

G06F 17/30; G06F 7/00

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

Korean utility models and applications for utility models

Japanese utility models and applications for utility models

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

eKOMPASS(KIPO internal) & Keywords: automatic, query, sub-queries, metadata, attribute, result, combination, and similar terms.

C. DOCUMENTS CONSIDERED TO BE RELEVANT

Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
X	US 8,375,029 B2 (CUI, ZHAN et al.) 12 February 2013 See column 3, lines 14-16, 34-53, 61-64; column 13, line 16 - column 14, line 2; column 18, lines 4-11; and claim 10.	1,6,11
Y		2-5,7-10,12-15
Y	US 2013-0311442 A1 (WEBER, THOMAS P. et al.) 21 November 2013 See paragraphs [0003], [0033], [0039], and [0042]; and figures 6 and 9.	2-5,7-10,12-15
A	US 8,655,867 B2 (COLLINS, JESSE et al.) 18 February 2014 See column 10, lines 23-28; column 11, lines 29-67; and claim 14.	1-15
A	US 7,437,349 B2 (BASU, SANKAR et al.) 14 October 2008 See column 6, line 50 - column 7, line 43; column 8, lines 4-39; column 9, lines 8-33; and figure 3.	1-15
A	US 2011-0258212 A1 (LU, JIANPING et al.) 20 October 2011 See paragraphs [0066]-[0070] and figure 2.	1-15



Further documents are listed in the continuation of Box C.



See patent family annex.

* Special categories of cited documents:

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

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

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

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

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

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

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

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

"&" document member of the same patent family

Date of the actual completion of the international search

10 February 2015 (10.02.2015)

Date of mailing of the international search report

11 February 2015 (11.02.2015)

Name and mailing address of the ISA/KR

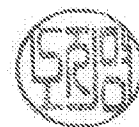
International Application Division
Korean Intellectual Property Office
189 Cheongsa-ro, Seo-gu, Daejeon Metropolitan City, 302-701,
Republic of Korea

Facsimile No. ++82 42 472 3473

Authorized officer

NHO, Ji Myong

Telephone No. +82-42-481-8528



INTERNATIONAL SEARCH REPORT

Information on patent family members

International application No.

PCT/US2014/044932

Patent document cited in search report	Publication date	Patent family member(s)	Publication date
US 8375029 B2	12/02/2013	EP 1918827 A1 EP 2078267 A1 US 2010-0070500 A1 WO 2008-053212 A1	07/05/2008 15/07/2009 18/03/2010 08/05/2008
US 2013-0311442 A1	21/11/2013	US 8825633 B2	02/09/2014
US 8655867 B2	18/02/2014	US 2011-0282864 A1 US 2014-0164357 A1	17/11/2011 12/06/2014
US 7437349 B2	14/10/2008	US 2003-0212666 A1	13/11/2003
US 2011-0258212 A1	20/10/2011	CN 102859523 A EP 2558965 A2 WO 2011-130008 A2 WO 2011-130008 A3	02/01/2013 20/02/2013 20/10/2011 12/01/2012