(12) INTERNATIONAL APPLICATION PUBLISHED UNDER THE PATENT COOPERATION TREATY (PCT)

(19) World Intellectual Property Organization
International Bureau

(43) International Publication Date
29 September 2016 (29.09.2016)

WIPO | PCT

(10) International Publication Number
WO 2016/153469 A1

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

(21) **International Application Number**:
PCT/US2015/021856

(22) **International Filing Date**:
20 March 2015 (20.03.2015)

(25) **Filing Language**: English

(26) **Publication Language**: English

(71) **Applicant**: **HEWLETT PACKARD ENTERPRISE DEVELOPMENT LP** [US/US]; 11445 Compaq Center Drive West, Houston, TX 77070 (US).

(72) **Inventors**: **CHEN, Qiming**; 1140 Enterprise Way, Sunnyvale, California 94089 (US). **HSU, Meichun**; 1140 Enterprise Way, Sunnyvale, California 94089 (US). **LIU, Rui**; 1140 Enterprise Way, Sunnyvale, California 94089 (US). **CASTELLANOS, Maria Guadalupe**; 1140 Enterprise Way, Sunnyvale, California 94089 (US).

(74) **Agents**: **KUIPERS, Luke W.** et al.; Hewlett Packard Enterprise, 3404 E. Harmony Road, Mail Stop 79, Fort Collins, CO 80528 (US).

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

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

Declarations under Rule 4.17:

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

[Continued on next page]

(54) **Title**: BASE USER DEFINED FUNCTIONS

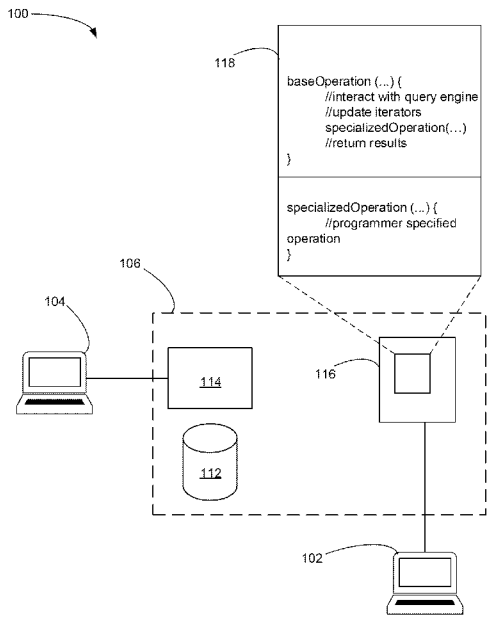

FIG. 1

(57) **Abstract**: Methods, devices, and techniques for base user defined functions in a database management system are discussed herein. For example, in one aspect, a query request is received from a computer device. The query request may include a query operator representing a specialized user defined function (SUDF). The SUDF may then be executed. Executing the SUDF may include executing a base operation of a base user defined function (BUDF). The base operation may interact with an application programming interface (API) of the query engine to obtain a tuple stored in the database. Executing the SUDF may further include executing a specialized operation that processes the tuple according to an analytics function. The specialized operation may generate a result. Then, a query result may be returned to the computer device. The query result can include the result.

WO 2016/153469 A1

Published:

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

BASE USER DEFINED FUNCTIONS

Background

[0001] A database management system (DBMS) is a system that manages databases and is capable of storing and retrieving large volumes of data. A Large scale DBMS can be implemented to support thousands of users accessing the databases via a wide assortment of applications. A DBMS can be structured to support a variety of different types of operations for a requesting entity (e.g., an application, the operating system or an end user). Such operations can be configured to retrieve, add, modify and delete data being stored and managed by the DBMS. Database access methods can support these operations using high-level query languages, such as the Structured Query Language (SQL). One of the primary operations performed with SQL is querying (also referred to herein as retrieving or selecting) data from data structures within a database.

Brief Description of the Drawings

[0002] FIG.1 is a diagram illustrating a system utilizing a base user defined function approach, according to an example.

[0003] FIG. 2 is a flowchart illustrating a method for loading a specialized user defined function within a base user defined function enabled database management system, according to an example.

[0004] FIG. 3 is a flowchart illustrating executing of a specialized user defined function, according to an example.

[0005] FIG. 4 is a block diagram illustrating a computer device, in accordance with an example

Detailed Description

[0008] Examples discussed herein may be applied to user defined functions (UDFs). A UDF may provide a mechanism for extending functionality of a database server by adding a function that can be evaluated in a query. A UDF that can be used as a relational operator in a SQL query is one way to add analytics functions to SQL queries. UDFs can also be used as a data source function for retrieving data from external systems. However, developing a UDF by directly interfacing with the APIs of a DBMS can be cumbersome and hard to maintain. As a result, wrapping each operator individually into a UDF can be tedious and may not be scalable when there is a large library of functions to be wrapped.

[0009] Examples discussed herein may address some of these issues by using a base UDF (BUDF) approach. A BUDF may be a base class definition (e.g., a Java class, C++, or any other suitable coding language) that encapsulates query engine interactions used by a UDF. Query engine interactions that may be encapsulated by the BUDF may include interactions for obtaining a host name, writing tuples, resolving a role, resolving a resource, and other suitable query engine interactions. Further, the BUDF may also specify various methods to be overridden by subclasses that specialize the BUDF. Such methods may define a developer supplied function (e.g., an analytics function) that operates on one or more tuples stored in the database. Accordingly, in some examples discussed herein, a developer may define a specialized user defined function (SUDF) that specializes the BUDF and overrides an operation (e.g., a pure virtual operation) defined by the BUDF. The overridden operation may include the user specific operation details, while the BUDF may hide the read/write interfaces of the database management system.

[0010] In one aspect discussed in the foregoing, a query request is received from a computer device. The query request may include a query operator representing a specialized user defined function (SUDF). The SUDF may then be executed. Executing the SUDF may include executing a base operation of a base user defined function (BUDF). The base operation may interact with an application programming

interface (API) of the query engine to obtain a tuple stored in the database. Executing the SUDF may further include executing a specialized operation that processes the tuple according to an analytics function. The specialized operation may generate a result. Then, a query result may be returned to the computer device. The query result can include the result.

[0011] The figures are now discussed in greater detail. FIG.1 is a diagram illustrating a system 100 utilizing a BUDF approach, according to an example. For example, the system 100 may include a UDF input machine 102, a querying machine 104, and a BUDF enabled database management system 106. The UDF input machine 102, the querying machine 104, and the BUDF enabled database management system 106 may be communicatively coupled through, for example, a network.

[0012] The UDF input machine 102 may be a computer system (e.g., one or more computer devices, such as desktops, laptops, networking devices, tablets, mobile phones, set-top boxes, and the like) that is configured to send, transmit, update, or delete user defined functions stored in the BUDF enabled database management system 106. In some cases, the UDF input machine 102 may be used by a database administrator or developer (not shown in FIG. 1) to define a UDF that specializes a BUDF. A UDF that specializes a BUDF may be referred to as a specializing UDF (SUDF). A UDF can specialize the BUDF through the use of facilities offered through programming languages such as C++, JAVA, C#, or any other suitable object oriented programming language.

[0013] The querying machine 104 may be a computer system (e.g., one or more computer devices, such as desktops, laptops, network devices, tablets, mobile phones, set-top boxes, and the like) that is configured to send query requests to the BUDF enabled database management system 106. The query requests may include data representing data expressed by a determinable syntax, such as structured query language (SQL), which may, in turn, include a reference to a SUDF.

[0014] The BUDF enabled database management system 106 is a computer system (e.g., one or more computer devices, such as desktops, laptops, network devices, tablets, mobile phones, set-top boxes, and the like) that is configured to receive UDFs from the UDF input machine 102, receive query requests from the querying machine 104, and execute the query requests according to the UDFs. In accordance with some examples, the BUDF enabled database management system 106 may include a database 112, a query engine 114, and a UDF manager 116. The database 112 may be a computer system configured to store searchable and identifiable data. In some cases, the data may be organized as tables and can be searchable using indexes.

[0015] The query engine 114 may be a module executed by a computer system for receiving and processing a query requests. In processing a query requests, the query engine 114 may parse the query requests to identify operators, operations, and UDFs expressed therein and coordinate execution of these operators, operations, and UDFs on the database to produce a query response.

[0016] The UDF manager 116 may be a module executed by a computer system for receiving UDFs from the UDF input device 102 and loading the UDFs, or otherwise making them accessible, to the query engine 114. FIG. 1 shows that the UDF manager 116 may store a SUDF 118. A SUDF may be a UDF that specializes a BUDF. Specialization may be achieved, in some cases, based on inheritance, as may be provided by a programming language, such as C++ or JAVA. The BUDF may include a base operation that interacts with the query engine and/or advances read/write interfaces (e.g., input/output iterators) that read and write from tables stored in the database 112. The advancement of the read/write operation may occur in the base operation that executes a partition execution loop. A partition execution loop may refer to herein code that is executed in a loop fashion so that each tuple in a table is processed. The base operation may call a specialized operation defined by the SUDF to perform an operation on current data (tuples pointed to by the

current read/write interfaces) from the tables stored in the database 112. The SUDF may take the form of executable codes, such as compiled code or a Java object.

[0017] Example operational aspects of a computer device are now discussed in greater detail.

[0018] FIG. 2 is a flowchart illustrating a method 200 for loading a SUDF within a BUDF enabled database management system, according to an example. The method 200 may be performed by the modules, components, systems shown in FIG. 1, and, accordingly, is described herein merely by way of reference thereto. For example, in some cases, the method 200 may be performed by a BUDF enabled database management system or, more precisely, in some cases, a UDF manager. It will be appreciated that the method 200 may, however, be performed on any suitable hardware.

[0019] The method 200 may begin when, at block 202, the UDF manager may receive a definition of a SUDF from a computer device. The SUDF may specialize a BUDF that includes: (a) a base operation that handles interactions with an API of a query engine of a database management system; and (b) a call to a specialized operation defined by the definition of the SUDF. The specialized operation may include instructions for processing a tuple according to an analytics function. Further, the specialized operation may include instructions for returning a result to the base operation.

[0020] At block, 204, the UDF manager may cause the SUDF to be available for execution by the query engine. In some cases, making the SUDF available to the query engine may involve a factory class (or instance thereof). For example, a specialized factory class may be defined that implements a loading operation that tells the query engine the name assigned to the SUDF (e.g., a class name) and the parameters and return values for the processing operation of the SUDF.

[0021] Once the UDF manager causes the SUDF to be available for execution, the query engine may then execute the SUDF when a corresponding query request

is received. A corresponding query request may refer to a query request that includes a relational operator, for example, that references the SUDF.

[0022] FIG. 3 is a flowchart illustrating executing of a SUDF, according to an example. The method 300 may be performed by the modules, components, systems shown in FIG. 1, and, accordingly, is described herein merely by way of reference thereto. For example, in some cases, the method 300 may be performed by a BUDF enabled database management system or, more precisely, in some cases, a query engine. It will be appreciated that the method 300 may, however, be performed on any suitable hardware.

[0023] The method 300 may begin at block 302 when the query engine receives a query request from a computer device (e.g., a querying machine, such as the querying machine 104 of FIG. 1). The query request may include a query operator representing a SUDF. For example, the query request may be in the form of a SQL request, where the SUDF is a query operator within the SQL request.

[0024] At block 304, the query engine may execute the SUDF. In one case, in executing the SUDF, the query engine may execute a base operation of a BUDF that interacts with an API of the query engine to obtain a tuple stored in the database (block 304A). Additionally or alternatively, in executing the SUDF, the query engine may execute a specialized operation that processes the tuple according to an analytics function (block 304B). The specialized operation being defined by a sub-class of the base user defined function, and the specialized operation may generate a result.

[0025] At block 306, the query engine may return a query result to the computer device. The query result may include the result generated by the specialized operation.

[0026] In some cases, the use of a BUDF makes developing and wrapping UDFs easier. Such may be the case because the explicit code needed to interact with the query engine is replaced with declarative statements. That is, the BUDF and factory classes can be used to generate the function signature automatically from a

developer's specification, and the BUDF can provide the code to cover the system interactions (e.g. read input/write output).

[0027] A BUDF can be reused for multiple SUDFs. For example, a SUDF that performs a first type of analytic operation may specialize the BUDF and another SUDF that performs a second type of analytic operation may likewise specialize the BUDF. In addition, a set of platform specific BUDFs may be provided such that the platform specific BUDF include logic to interact with different database platforms, such as a Graph DB or HDFS. In this way a UDF hierarchy can be formed, and by inheriting from the prebuilt BUDFs, the development of a specific SUDF can be greatly simplified.

[0028] FIG. 4 is a block diagram illustrating a computer device 400, in accordance with an example. The computer device 400 may include a processor 441 and a computer-readable storage device 442. The processor 441 may be a device suitable to read and execute processor executable instructions, such as a CPU, or an integrated circuit configured to perform a configured function. The processor executable instructions may cause the processor 441 to implement techniques described herein. For example the processor 441 may execute instructions for enabling a BUDF in a database management system. For example, the processor may receive a definition of a SUDF from a computer device. The SUDF may specialize a BUDF that includes: (a) a base operation includes a processing block that advances a read iterator to a database table, and a call to a specialized operation defined by the definition of the SUDF. The specialized operation processes a tuple according to an analytics function and returns a result to the base operation. The processor is also to expose the SUDF as an executable operator for the query engine

[0029] The processor 441 shown in FIG. 4 is coupled to the computer-readable storage device 442. The computer-readable storage device 442 may contain thereon a set of instructions, which when executed by the processor 441, cause the processor 441 to execute the techniques described herein. For example, the

computer-readable storage device 442 may include SUDF executable instructions 444. Further, the computer-readable storage device 442 may include a SUDF object 446. The SUDF object 446 may be data and/or logic that is usable to execute a SUDF.

[0030] For example, in one aspect, execution of the instructions 444, whole or in part, may cause the processor 441 to receive a definition of a specialized user defined function (SUDF) (e.g., the SUDF object 446) from a computer device. The SUDF may specialize a base user defined function (BUDF). The BUDF can include: (a) a base operation that handles interactions with an application programming interface (API) of a query engine of a database management system, and (b) a call to a specialized operation defined by the definition of the SUDF. The specialized operation processes a tuple according to an analytics function and returns a result to the base operation. Execution of the instructions can further cause processor to cause the SUDF to be available for execution by the query engine.

Claims

What is claimed is:

1. A method comprising:

receiving, by at least one processor, a query request from a computer device, the query request including a query operator representing a specialized user defined function (SUDF) that specializes a base user defined function (BUDF);

executing, by the at least one processor, the SUDF, executing the SUDF comprises:

executing a base operation of the BUDF that interacts with an application programming interface (API) of the query engine to obtain a tuple stored in the database;

executing a specialized operation that processes the tuple according to an analytics function, the specialized operation being defined by the SUDF, the specialized operation generating a result; and

returning, by the at least one processor, a query result to the computer device, the query result including the result.

2. The method of claim 1, wherein the base operation of the BUDF obtains the tuple stored in the database through a read interface.

3. The method of claim 2, wherein the base operation of the BUDF calls the specialized operation in a partition execution loop that advances the read interface at each iteration of the partition execution loop.

4. The method of claim 1, wherein the BUDF connects to a given database platform.

5. The method of claim 1, wherein the SUDF is a Java executable object.

6. A device comprising:

a processor; and

a machine-readable storage device comprising instructions that, when executed, cause the processor to:

receive a definition of a specialized user defined function (SUDF) from a computer device, the SUDF specializes a base user defined function (BUDF) that includes: a base operation includes a processing block that advances a read iterator to a database table, and a call to a specialized operation defined by the definition of the SUDF using a tuple pointed to by the read iterator, wherein the specialized operation processes the tuple according to an analytics function and returns a result to the base operation; and

exposing the SUDF as an executable operator for the query engine.

7. The device of claim 6, wherein the base operation of the BUDF obtains the tuple stored in the database through a read interface.

8. The method of claim 7, wherein the base operation of the BUDF calls the specialized operation in a partition execution loop that advances the read interface at each iteration of the partition execution loop.

9. The method of claim 7, wherein the BUDF includes API call to connect to a given database platform.

10. The method of claim 6, wherein the SUDF is an executable Java object.

11. A machine-readable storage device comprising instructions that, when executed, cause a processor to:

receive a definition of a specialized user defined function (SUDF) from a computer device, the SUDF specializes a base user defined function (BUDF) that includes: a base operation that handles interactions with an application programming interface (API) of a query engine of a database management system, and a call to a specialized operation defined by the definition of the SUDF, wherein the specialized operation processes a tuple according to an analytics function and returns a result to the base operation; and

cause the SUDF to be available for execution by the query engine.

12. The machine-readable storage device of claim 11, wherein the tuple is passed to the specialized operation via the base operation.

13. The machine-readable storage device of claim 11, wherein SUDF is a Java executable object.

14. The machine-readable storage device of claim 11, wherein the BUDF declares the special operation and the SUDF overrides the special operation with a first analytic function.

15. The machine-readable storage device of claim 14, wherein the instructions that, when executed, further cause the processor to:

receive an additional definition of an additional SUDF from the computer device, the additional SUDF specializes the BUDF, wherein the additional SUDF overrides the specialized operation declared by the BUDF with a second analytic function.

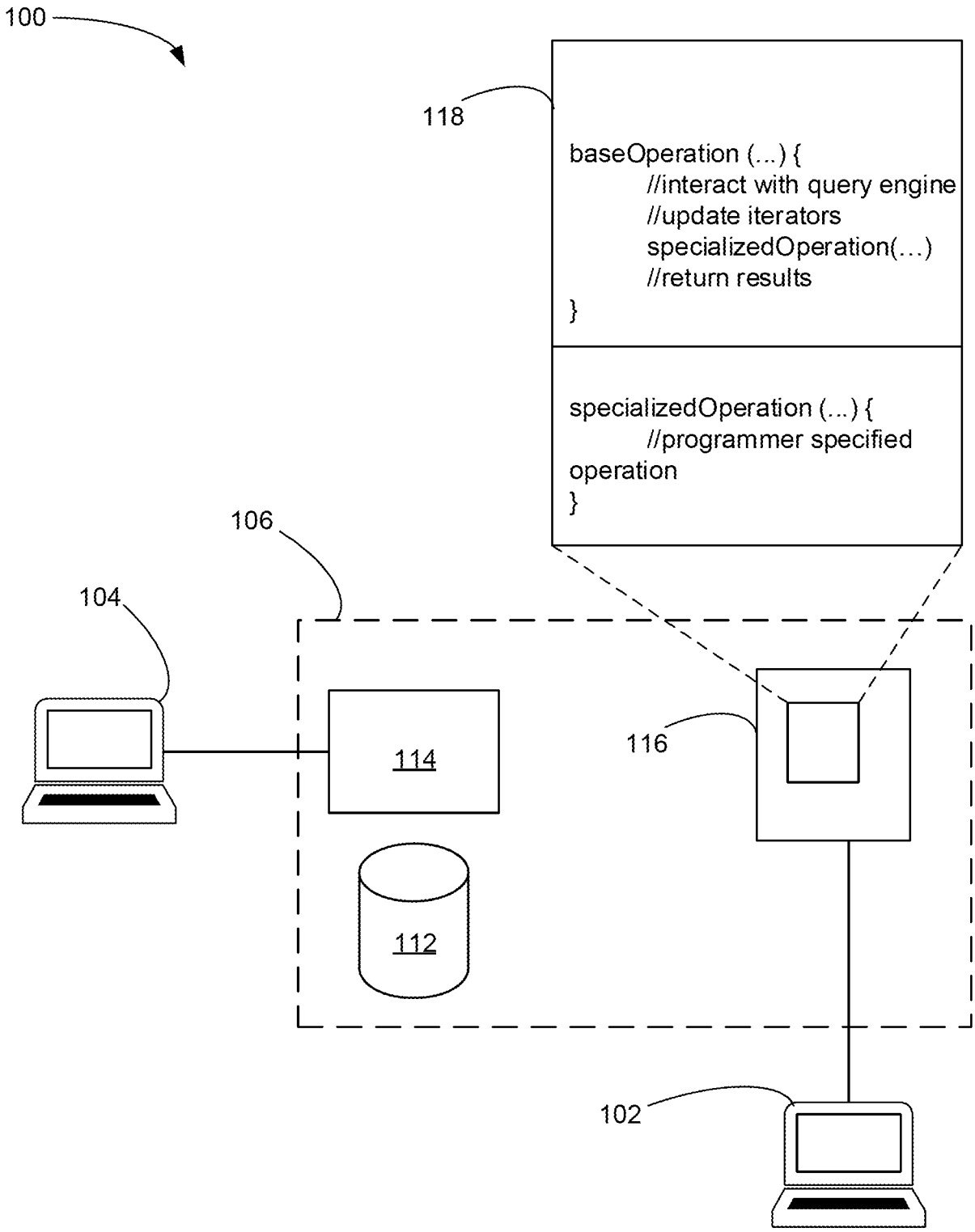
100
118
baseOperation (...) {
//interact with query engine
//update iterators
specializedOperation(...)
//return results
}
specializedOperation (...) {
//programmer specified
operation
}
106
104
114
116
112
102

FIG. 1

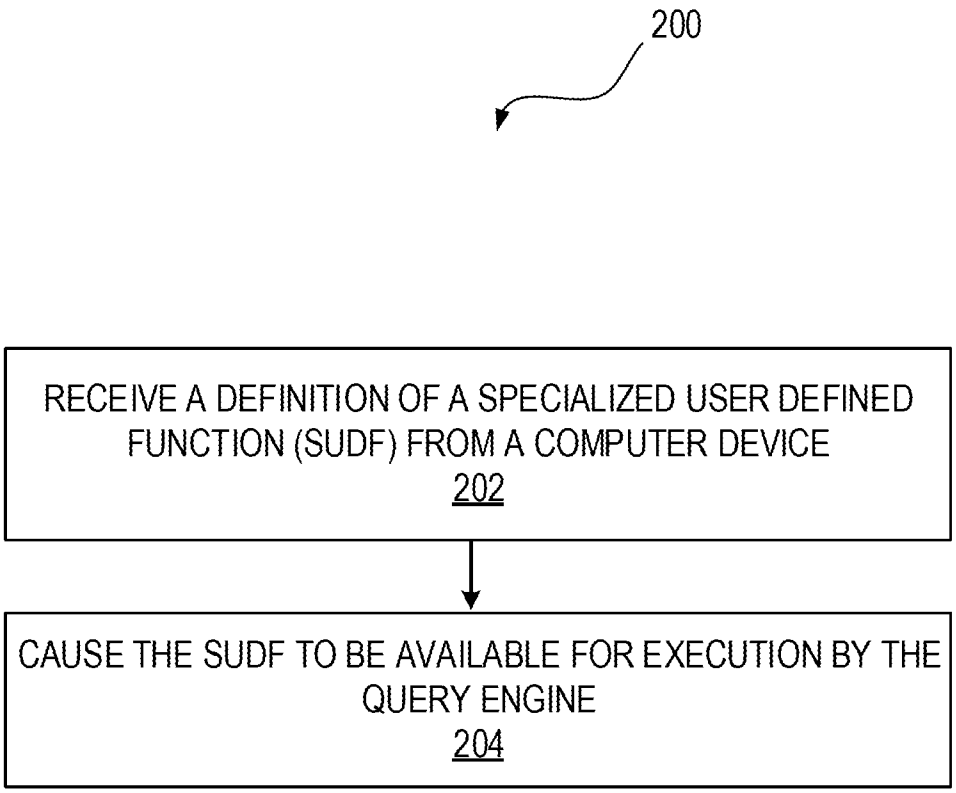
200
RECEIVE A DEFINITION OF A SPECIALIZED USER DEFINED FUNCTION (SUDF) FROM A COMPUTER DEVICE
202
CAUSE THE SUDF TO BE AVAILABLE FOR EXECUTION BY THE QUERY ENGINE
204

FIG. 2

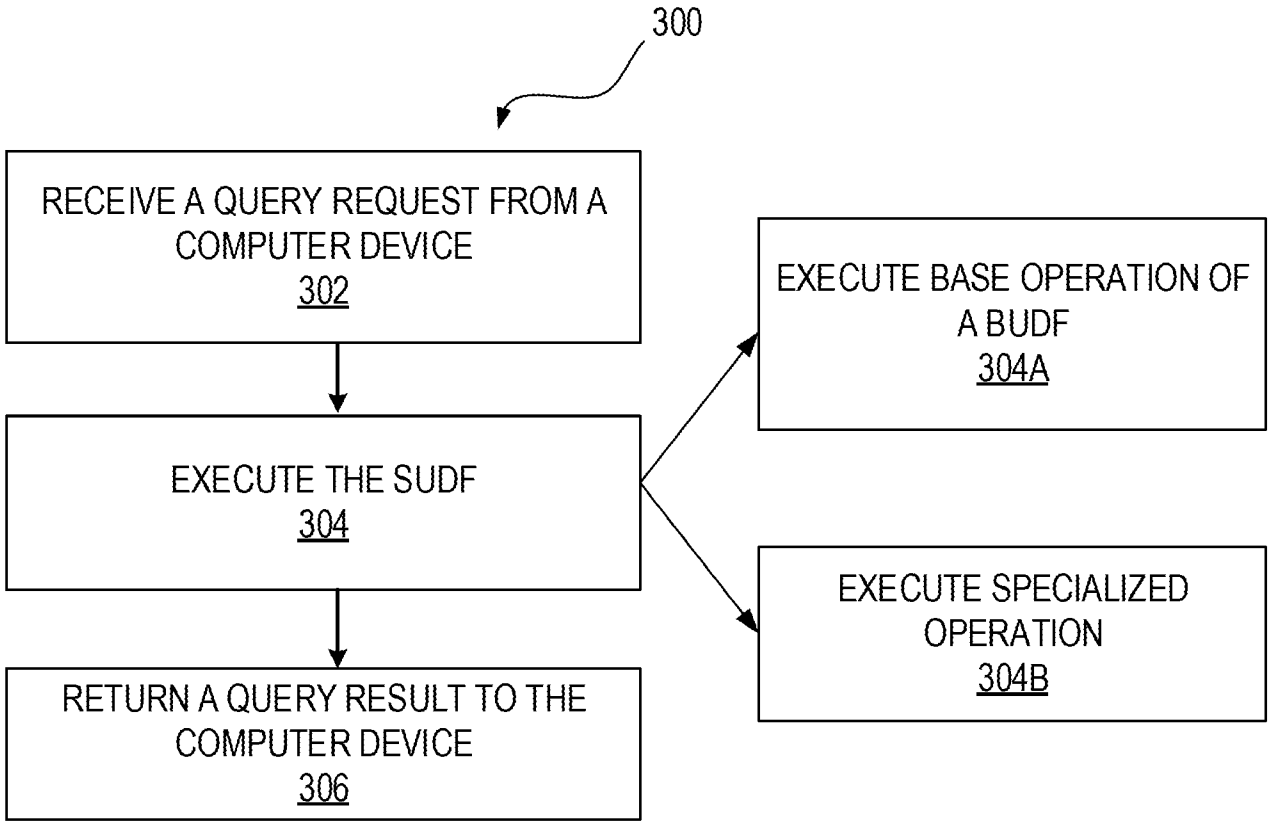
300
RECEIVE A QUERY REQUEST FROM A COMPUTER DEVICE
302
EXECUTE THE SUDF
304
RETURN A QUERY RESULT TO THE COMPUTER DEVICE
306
EXECUTE BASE OPERATION OF A BUDF
304A
EXECUTE SPECIALIZED OPERATION
304B

FIG. 3

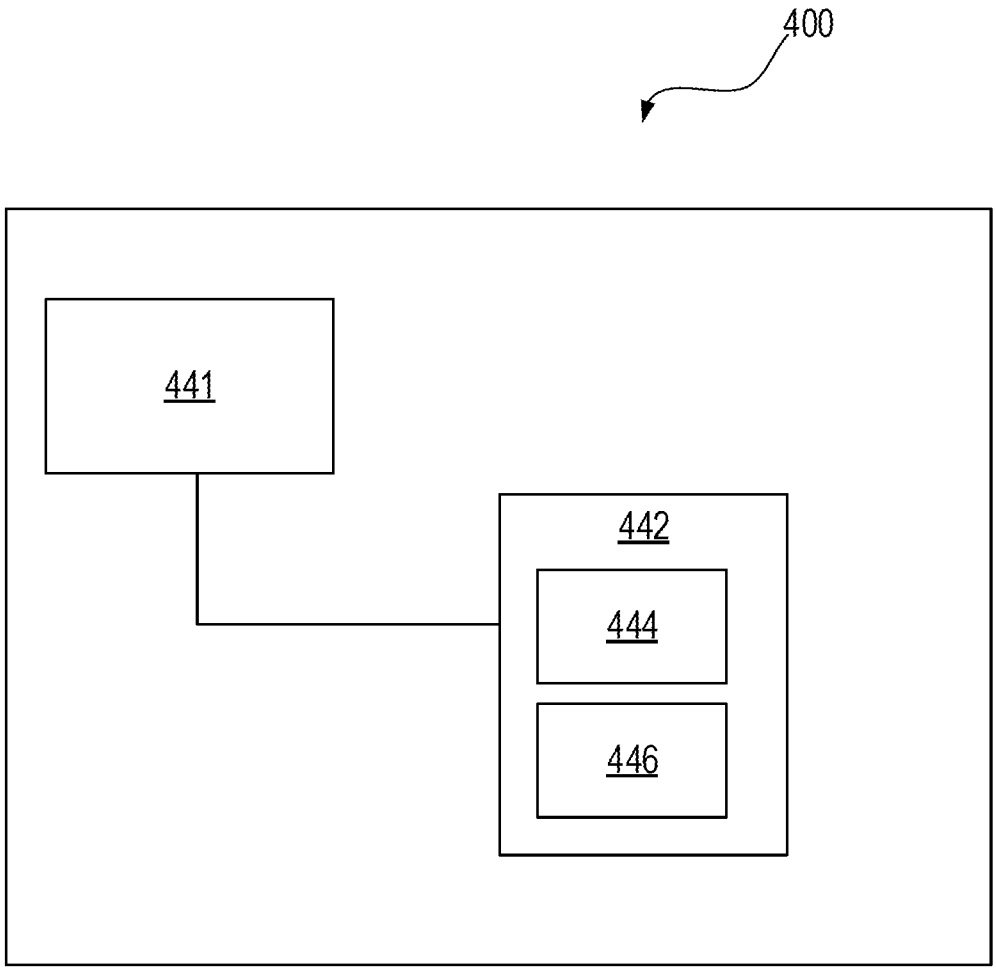

FIG. 4

INTERNATIONAL SEARCH REPORT

International application No.
PCT/US2015/021856

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; G06F 7/06

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: query, request, specialized, user defined function(UDF), returning, tuple, and similar terms.

C. DOCUMENTS CONSIDERED TO BE RELEVANT

Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
X	US 2012-0191732 A1 (MUTHIAN GEORGE) 26 July 2012 See paragraphs [0009]-[0012] and [0027]; claims 1-2, 7, and 13; and figure 1.	1-15
A	US 2008-0270392 A1 (PAUL REUBEN DAY et al.) 30 October 2008 See paragraphs [0051]-[0059]; and figures 5A-5B.	1-15
A	US 2009-0112799 A1 (ERIC LAWRENCE BARSNESS et al.) 30 April 2009 See paragraphs [0047]-[0049]; and figure 4.	1-15
A	US 2009-0043910 A1 (ERIC L BARSNESS et al.) 12 February 2009 See paragraphs [0068]-[0069]; and figures 8-9.	1-15
A	WO 2015-039072 A1 (MASTERCARD INTERNATIONAL INCORPORATED) 19 March 2015 See paragraphs [0055]-[0057]; and figures 4-5.	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	Date of mailing of the international search report
20 April 2016 (20.04.2016)	**22 April 2016 (22.04.2016)**

Name and mailing address of the ISA/KR
International Application Division
Korean Intellectual Property Office
189 Cheongsa-ro, Seo-gu, Daejeon Metropolitan City, 35208,
Republic of Korea
Facsimile No. +82-42-481-8578

Authorized officer
KIM, KI HO
Telephone No. +82-42-481-8691

INTERNATIONAL SEARCH REPORT

Information on patent family members

International application No.
PCT/US2015/021856

Patent document cited in search report	Publication date	Patent family member(s)	Publication date
US 2012-0191732 A1	26/07/2012	US 8856151 B2	07/10/2014
US 2008-0270392 A1	30/10/2008	US 2006-0026116 A1	02/02/2006
		US 2008-0275858 A1	06/11/2008
		US 7447680 B2	04/11/2008
		US 8041708 B2	18/10/2011
		US 8099410 B2	17/01/2012
US 2009-0112799 A1	30/04/2009	US 8682875 B2	25/03/2014
US 2009-0043910 A1	12/02/2009	TW 200921432 A	16/05/2009
		US 8171047 B2	01/05/2012
WO 2015-039072 A1	19/03/2015	US 2015-0081666 A1	19/03/2015