US 20100241893A1

(19) **United States**
(12) **Patent Application Publication**
Friedman et al.
(10) **Pub. No.: US 2010/0241893 A1**
(43) **Pub. Date: Sep. 23, 2010**

(54) **INTERPRETATION AND EXECUTION OF A CUSTOMIZABLE DATABASE REQUEST USING AN EXTENSIBLE COMPUTER PROCESS AND AN AVAILABLE COMPUTING ENVIRONMENT**

(76) Inventors: **ERIC Friedman**, Mountain View, CA (US); **Peter Pawlowski**, Menlo Park, CA (US)

Correspondence Address:
Raj Abhyanker, P.C.
1580 West, El Camino Real, Suite 8
Mountain View, CA 94040 (US)

(21) Appl. No.: **12/406,875**

(22) Filed: **Mar. 18, 2009**

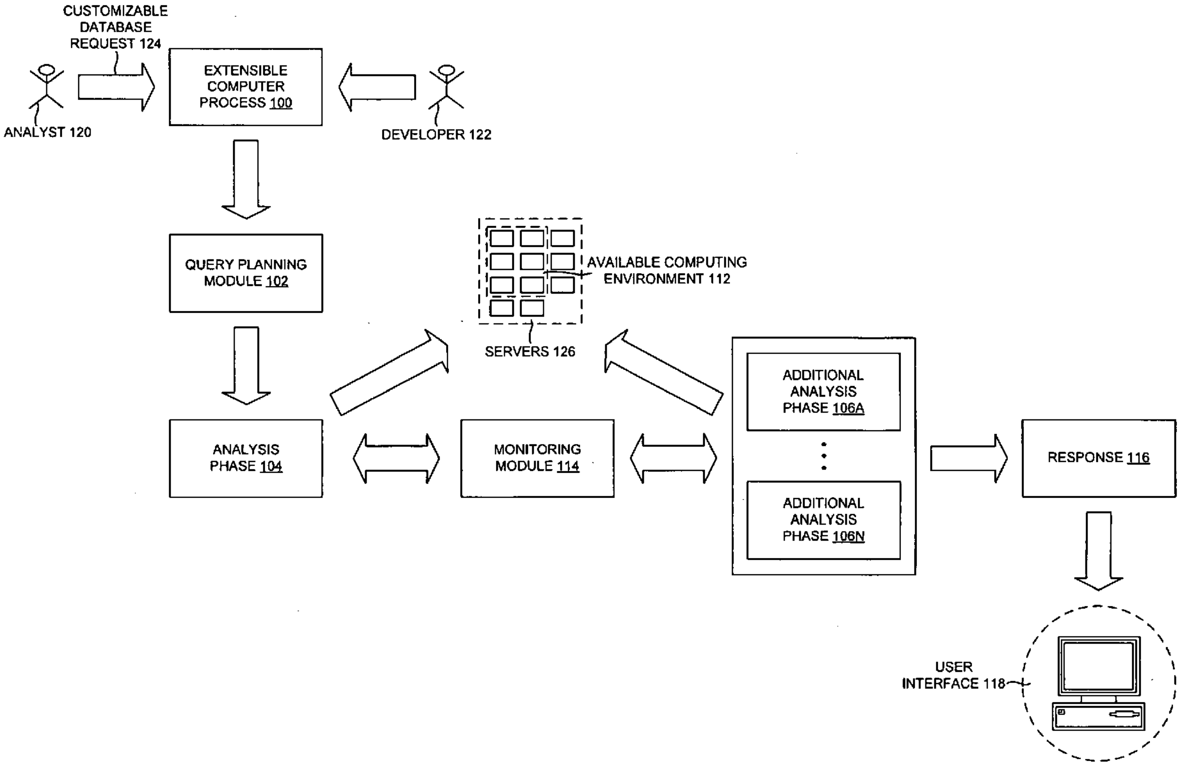
Publication Classification

(51) **Int. Cl.**
G06F 17/30 (2006.01)
G06F 11/07 (2006.01)

(52) **U.S. Cl.** **714/2**; 707/E17.009; 714/E11.023

(57) **ABSTRACT**

Interpretation and execution of a customizable database request using an extensible computer process and an available computing environment is disclosed. In an embodiment, a method includes generating an interpretation of a customizable database request which includes an extensible computer process and providing an input guidance to available processors of an available computing environment. The method further includes automatically distributing an execution of the interpretation across the available computing environment operating concurrently and in parallel, wherein a component of the execution is limited to at least a part of an input data. The method also includes automatically assembling a response using a distributed output of the execution.


CUSTOMIZABLE DATABASE REQUEST 124
ANALYST 120
EXTENSIBLE COMPUTER PROCESS 100
DEVELOPER 122
QUERY PLANNING MODULE 102
AVAILABLE COMPUTING ENVIRONMENT 112
SERVERS 126
ANALYSIS PHASE 104
MONITORING MODULE 114
ADDITIONAL ANALYSIS PHASE 106A
ADDITIONAL ANALYSIS PHASE 106N
RESPONSE 116
USER INTERFACE 118

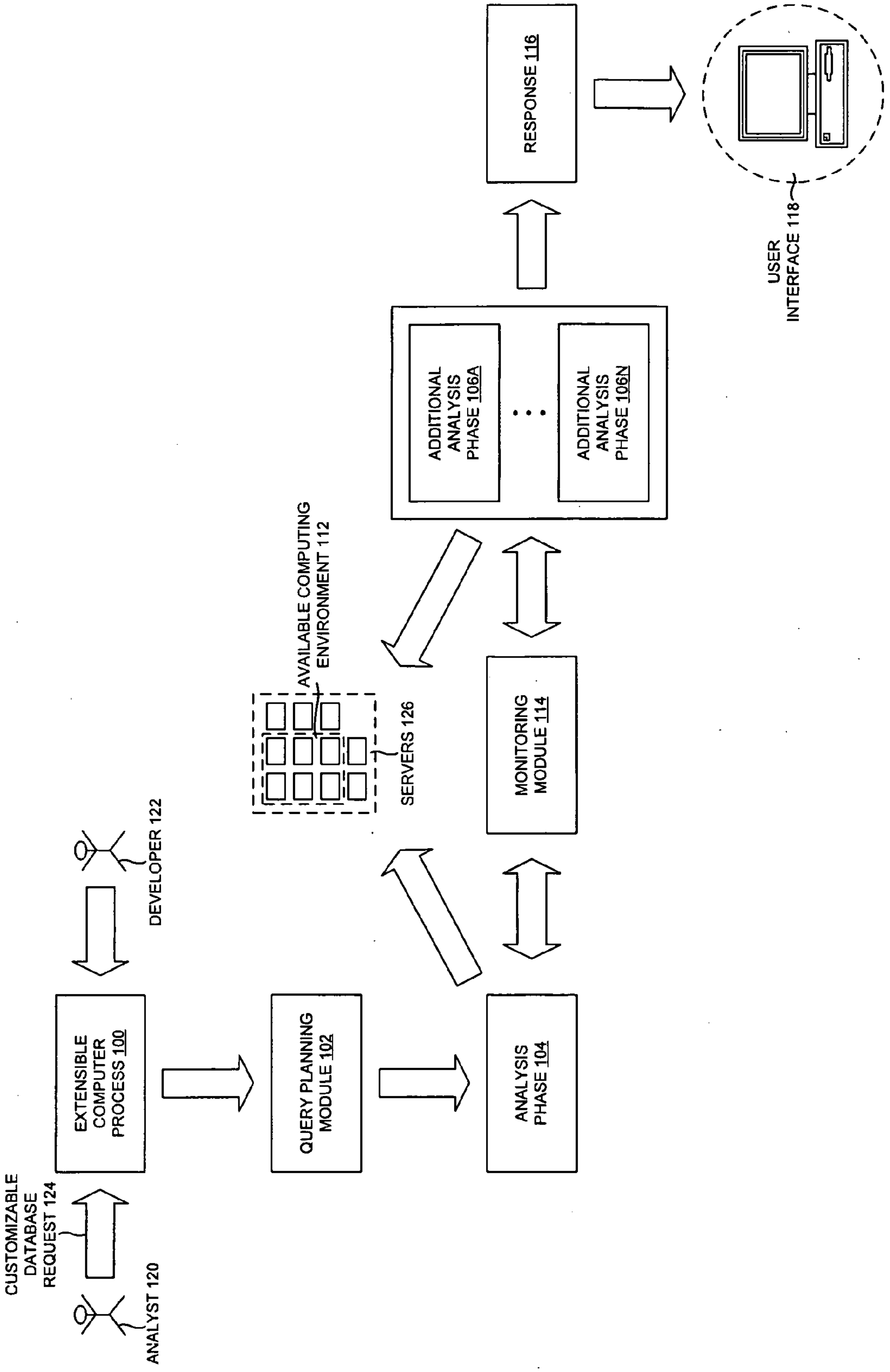

ANALYST 120
CUSTOMIZABLE DATABASE REQUEST 124
EXTENSIBLE COMPUTER PROCESS 100
DEVELOPER 122
QUERY PLANNING MODULE 102
ANALYSIS PHASE 104
SERVERS 126
AVAILABLE COMPUTING ENVIRONMENT 112
MONITORING MODULE 114
ADDITIONAL ANALYSIS PHASE 106A
ADDITIONAL ANALYSIS PHASE 106N
RESPONSE 116
USER INTERFACE 118

FIGURE 1

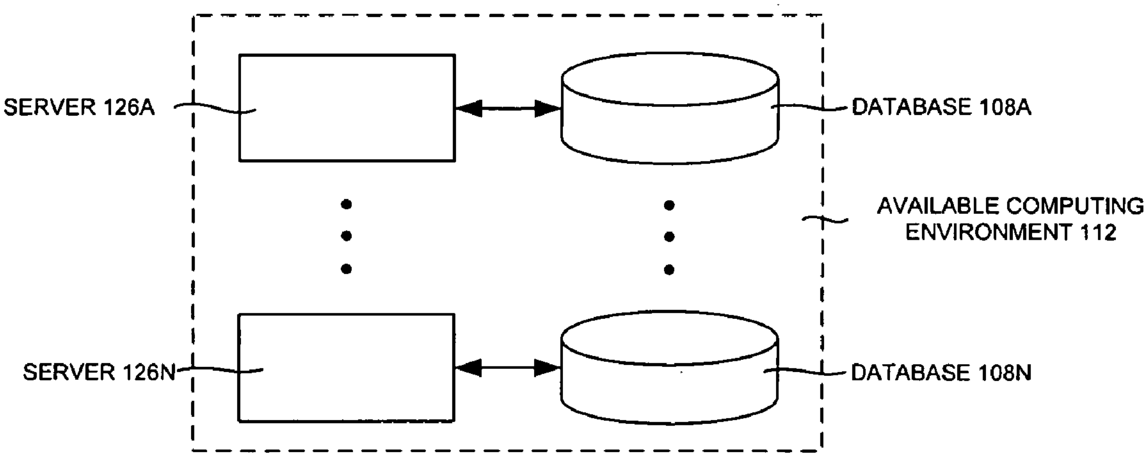

SERVER 126A
DATABASE 108A
AVAILABLE COMPUTING ENVIRONMENT 112
SERVER 126N
DATABASE 108N

FIGURE 2

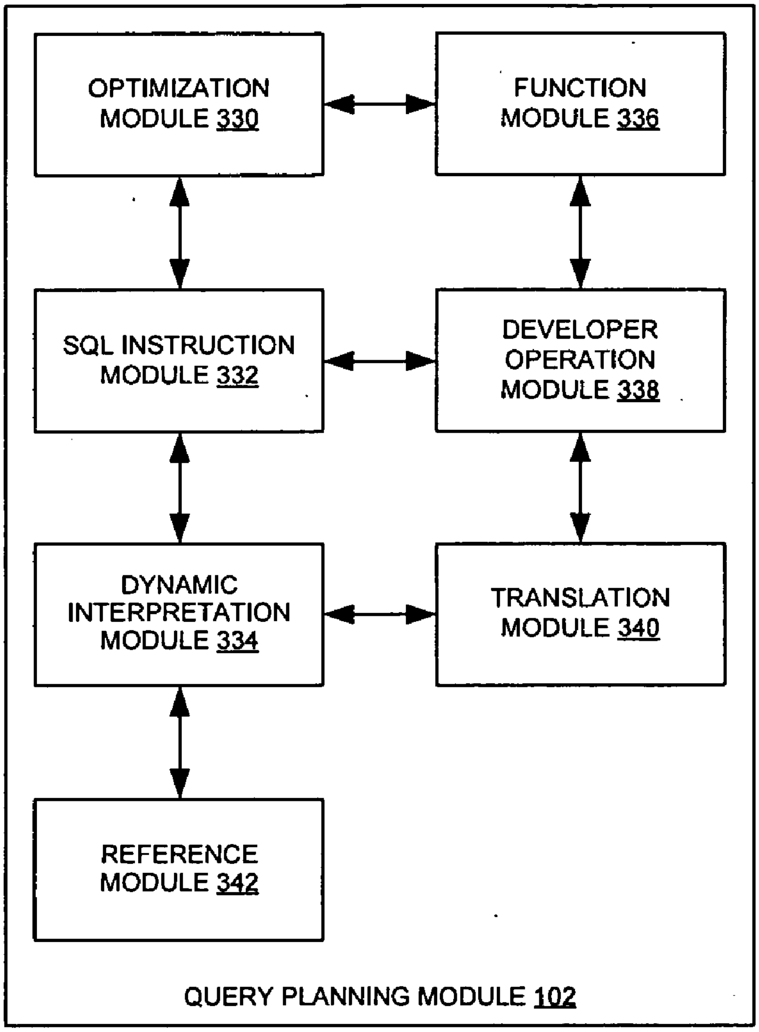

OPTIMIZATION MODULE 330
FUNCTION MODULE 336
SQL INSTRUCTION MODULE 332
DEVELOPER OPERATION MODULE 338
DYNAMIC INTERPRETATION MODULE 334
TRANSLATION MODULE 340
REFERENCE MODULE 342
QUERY PLANNING MODULE 102

FIGURE 3

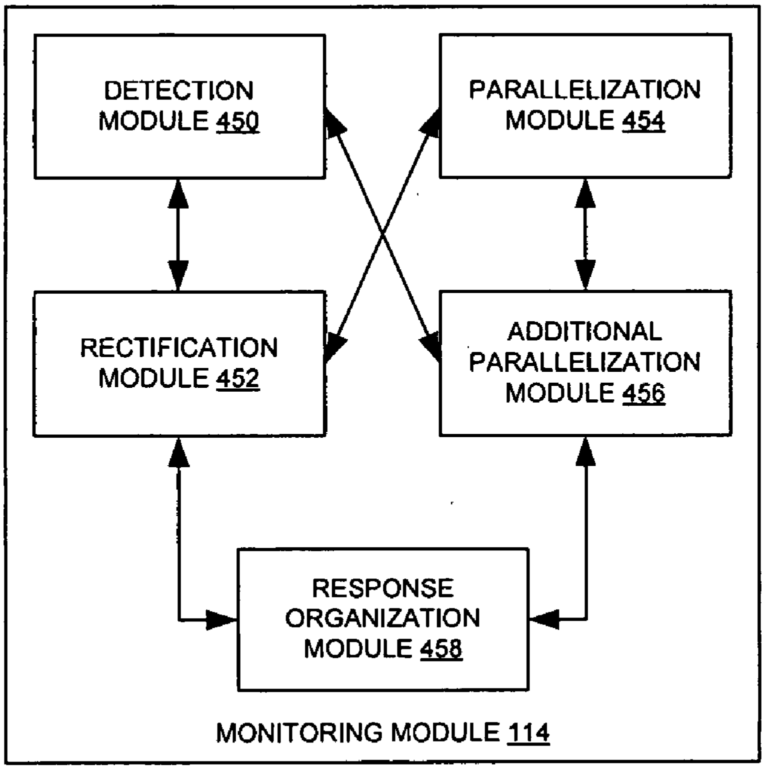

DETECTION MODULE 450
PARALLELIZATION MODULE 454
RECTIFICATION MODULE 452
ADDITIONAL PARALLELIZATION MODULE 456
RESPONSE ORGANIZATION MODULE 458
MONITORING MODULE 114

FIGURE 4

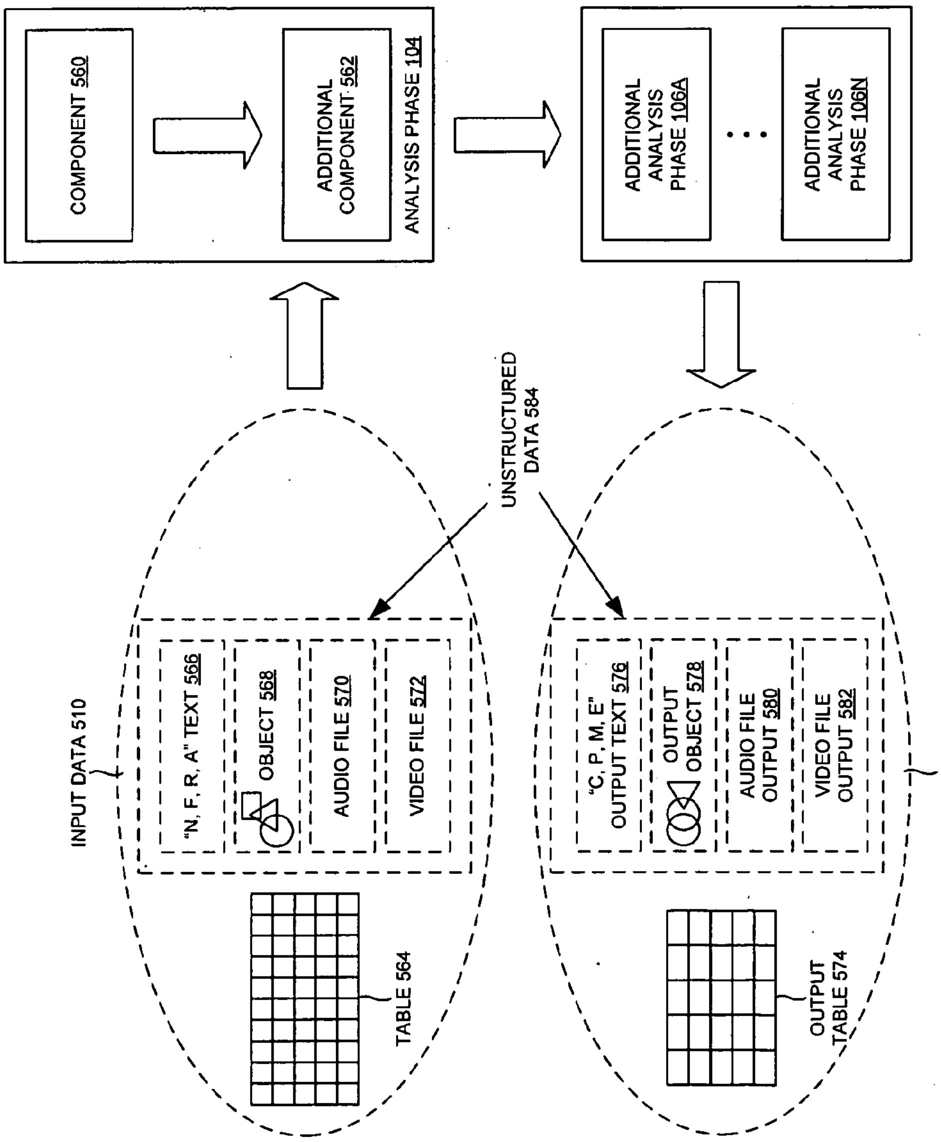

COMPONENT 560
ADDITIONAL COMPONENT 562
ANALYSIS PHASE 104
ADDITIONAL ANALYSIS PHASE 106A
ADDITIONAL ANALYSIS PHASE 106N
UNSTRUCTURED DATA 584
INPUT DATA 510
"N, F, R, A" TEXT 566
OBJECT 568
AUDIO FILE 570
VIDEO FILE 572
TABLE 564
RESPONSE 116
"C, P, M, E" OUTPUT TEXT 576
OUTPUT OBJECT 578
AUDIO FILE OUTPUT 580
VIDEO FILE OUTPUT 582
OUTPUT TABLE 574

FIGURE 5

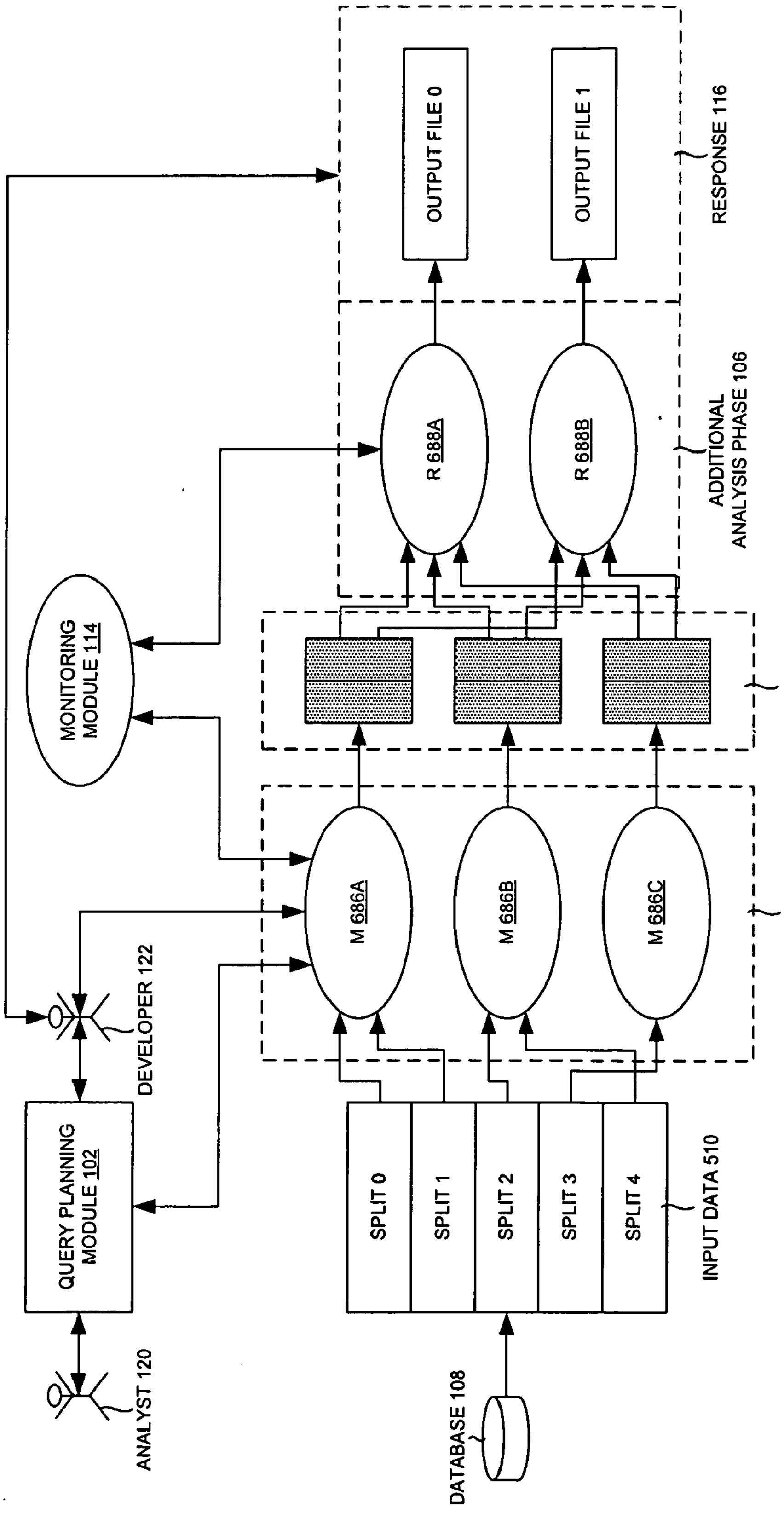

ANALYST 120
QUERY PLANNING MODULE 102
DEVELOPER 122
MONITORING MODULE 114
DATABASE 108
SPLIT 0
SPLIT 1
SPLIT 2
SPLIT 3
SPLIT 4
INPUT DATA 510
M 686A
M 686B
M 686C
ANALYSIS PHASE 104
INTERMEDIATE FILES 690
R 688A
R 688B
ADDITIONAL ANALYSIS PHASE 106
OUTPUT FILE 0
OUTPUT FILE 1
RESPONSE 116

FIGURE 6

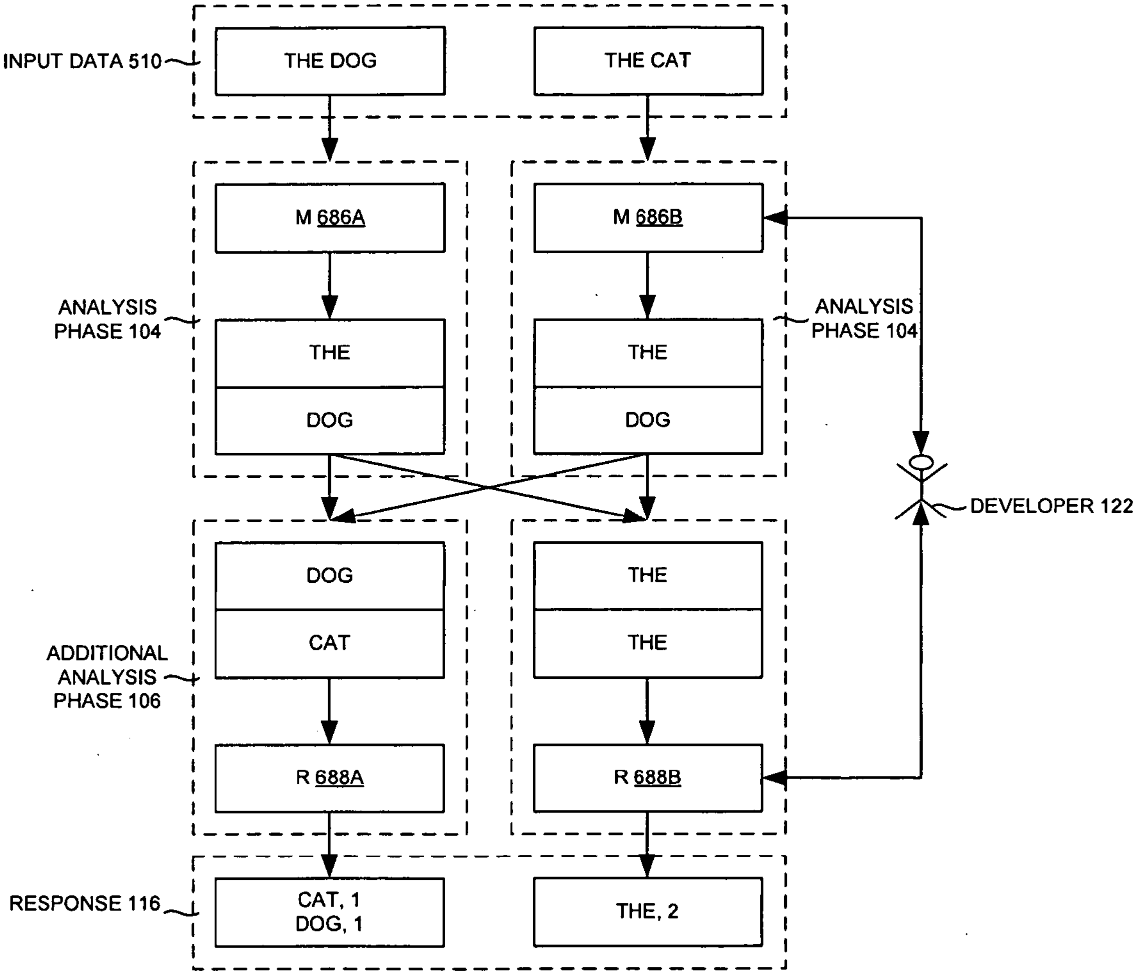

INPUT DATA 510
THE DOG
THE CAT
M 686A
M 686B
ANALYSIS PHASE 104
THE
DOG
ANALYSIS PHASE 104
THE
DOG
DEVELOPER 122
DOG
CAT
THE
THE
ADDITIONAL ANALYSIS PHASE 106
R 688A
R 688B
RESPONSE 116
CAT, 1
DOG, 1
THE, 2

FIGURE 7

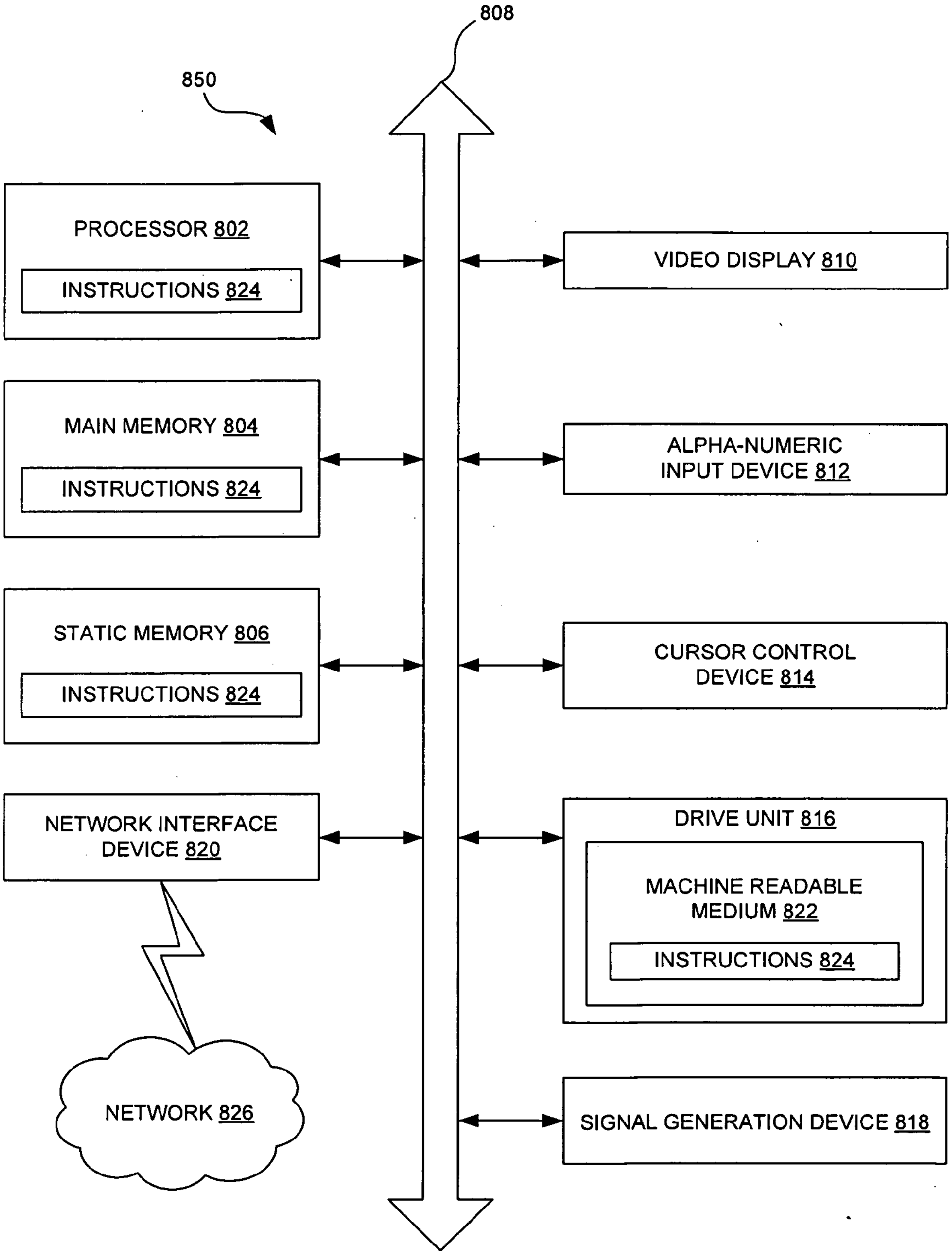

808
850
PROCESSOR 802
INSTRUCTIONS 824
MAIN MEMORY 804
INSTRUCTIONS 824
STATIC MEMORY 806
INSTRUCTIONS 824
NETWORK INTERFACE DEVICE 820
NETWORK 826
VIDEO DISPLAY 810
ALPHA-NUMERIC INPUT DEVICE 812
CURSOR CONTROL DEVICE 814
DRIVE UNIT 816
MACHINE READABLE MEDIUM 822
INSTRUCTIONS 824
SIGNAL GENERATION DEVICE 818

FIGURE 8

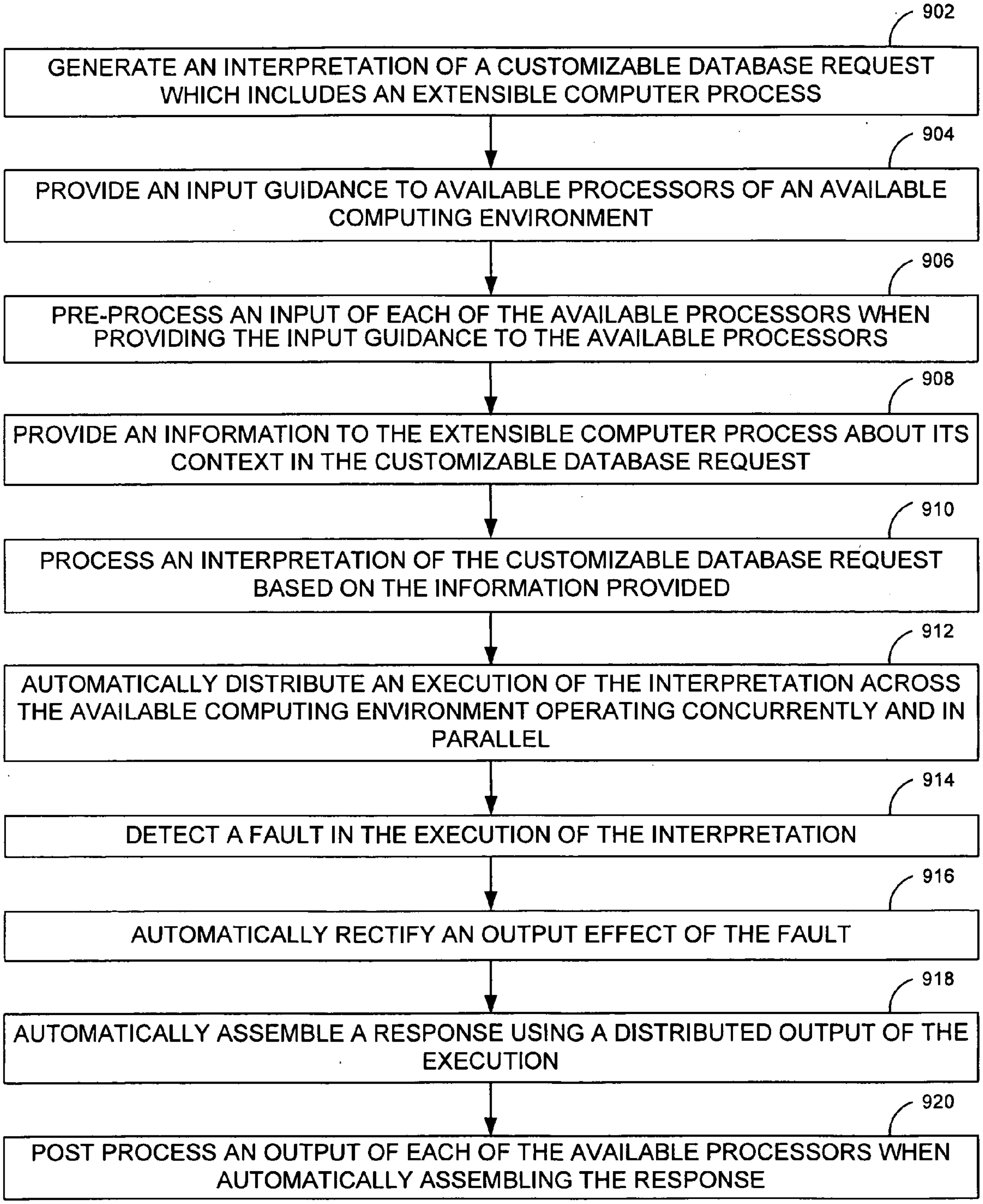

902
GENERATE AN INTERPRETATION OF A CUSTOMIZABLE DATABASE REQUEST WHICH INCLUDES AN EXTENSIBLE COMPUTER PROCESS
904
PROVIDE AN INPUT GUIDANCE TO AVAILABLE PROCESSORS OF AN AVAILABLE COMPUTING ENVIRONMENT
906
PRE-PROCESS AN INPUT OF EACH OF THE AVAILABLE PROCESSORS WHEN PROVIDING THE INPUT GUIDANCE TO THE AVAILABLE PROCESSORS
908
PROVIDE AN INFORMATION TO THE EXTENSIBLE COMPUTER PROCESS ABOUT ITS CONTEXT IN THE CUSTOMIZABLE DATABASE REQUEST
910
PROCESS AN INTERPRETATION OF THE CUSTOMIZABLE DATABASE REQUEST BASED ON THE INFORMATION PROVIDED
912
AUTOMATICALLY DISTRIBUTE AN EXECUTION OF THE INTERPRETATION ACROSS THE AVAILABLE COMPUTING ENVIRONMENT OPERATING CONCURRENTLY AND IN PARALLEL
914
DETECT A FAULT IN THE EXECUTION OF THE INTERPRETATION
916
AUTOMATICALLY RECTIFY AN OUTPUT EFFECT OF THE FAULT
918
AUTOMATICALLY ASSEMBLE A RESPONSE USING A DISTRIBUTED OUTPUT OF THE EXECUTION
920
POST PROCESS AN OUTPUT OF EACH OF THE AVAILABLE PROCESSORS WHEN AUTOMATICALLY ASSEMBLING THE RESPONSE

FIGURE 9

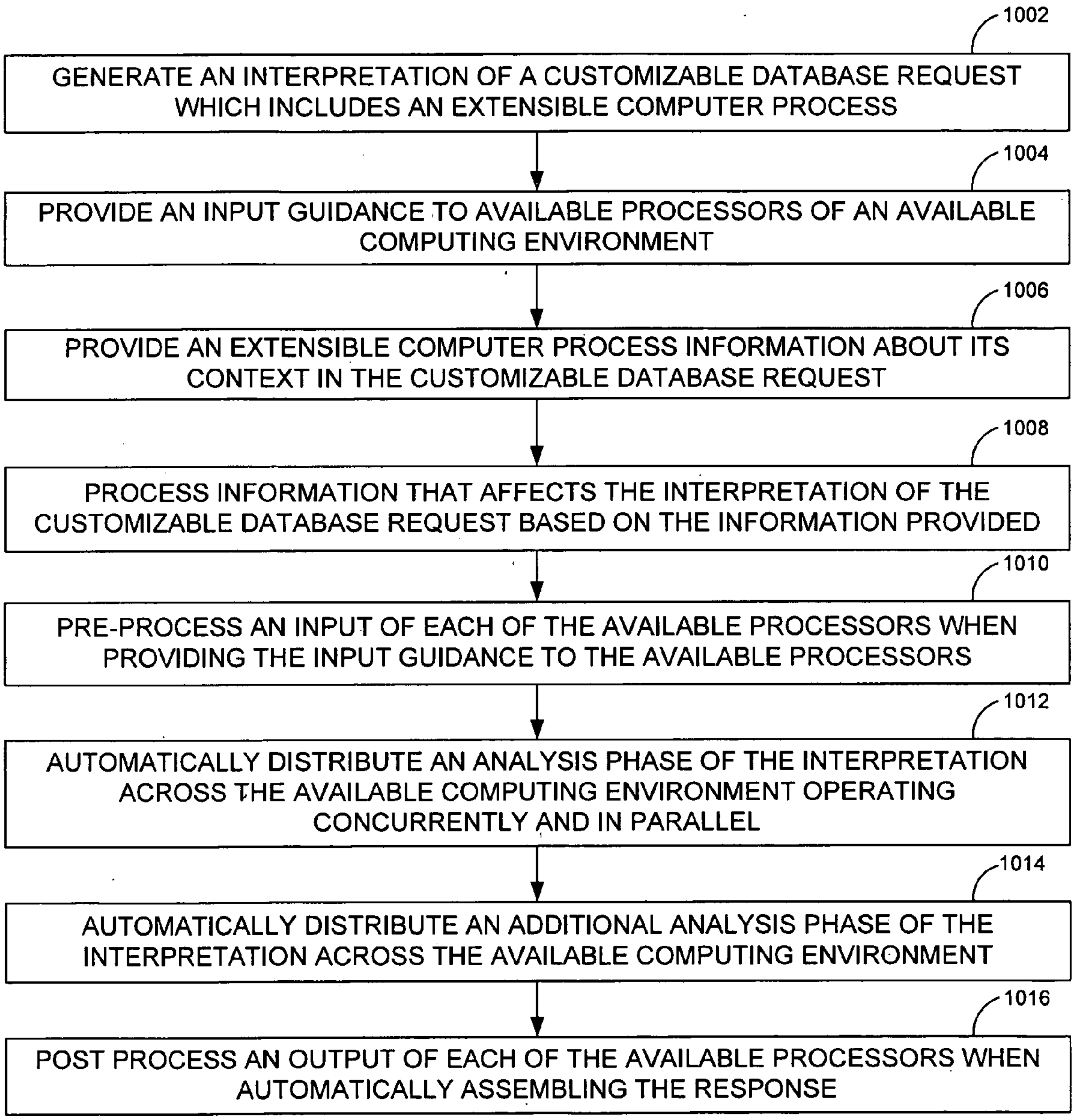

1002
GENERATE AN INTERPRETATION OF A CUSTOMIZABLE DATABASE REQUEST WHICH INCLUDES AN EXTENSIBLE COMPUTER PROCESS
1004
PROVIDE AN INPUT GUIDANCE TO AVAILABLE PROCESSORS OF AN AVAILABLE COMPUTING ENVIRONMENT
1006
PROVIDE AN EXTENSIBLE COMPUTER PROCESS INFORMATION ABOUT ITS CONTEXT IN THE CUSTOMIZABLE DATABASE REQUEST
1008
PROCESS INFORMATION THAT AFFECTS THE INTERPRETATION OF THE CUSTOMIZABLE DATABASE REQUEST BASED ON THE INFORMATION PROVIDED
1010
PRE-PROCESS AN INPUT OF EACH OF THE AVAILABLE PROCESSORS WHEN PROVIDING THE INPUT GUIDANCE TO THE AVAILABLE PROCESSORS
1012
AUTOMATICALLY DISTRIBUTE AN ANALYSIS PHASE OF THE INTERPRETATION ACROSS THE AVAILABLE COMPUTING ENVIRONMENT OPERATING CONCURRENTLY AND IN PARALLEL
1014
AUTOMATICALLY DISTRIBUTE AN ADDITIONAL ANALYSIS PHASE OF THE INTERPRETATION ACROSS THE AVAILABLE COMPUTING ENVIRONMENT
1016
POST PROCESS AN OUTPUT OF EACH OF THE AVAILABLE PROCESSORS WHEN AUTOMATICALLY ASSEMBLING THE RESPONSE

FIGURE 10

INTERPRETATION AND EXECUTION OF A CUSTOMIZABLE DATABASE REQUEST USING AN EXTENSIBLE COMPUTER PROCESS AND AN AVAILABLE COMPUTING ENVIRONMENT

FIELD OF TECHNOLOGY

[0001] This disclosure relates generally to interpretation and execution of a customizable database request using an extensible computer process and an available computing environment.

BACKGROUND

[0002] A database analyst may seek to request information from a database but may be prevented from doing so by a lack of an ability to customize a database query. The database analyst may also be unable to distribute the processing of the query across a distributed computational environment, which may include one or more servers. The database analyst may be restricted to a limited set of queries that may limit the effectiveness of the analyst's ability to obtain information from the database. The analyst may therefore seek data inefficiently using an excessive number of queries. The data analyst may also be required to transfer the processed information of the database to a separate process to analyze the data. The database analyst may therefore be required to spend an excessive amount of time obtaining information, which may lead to a delay, an additional cost of the analyst's time, an additional time for a processor usage, and a greater possibility of incurring a human made error. The database analyst may ultimately fail to find a desired information.

SUMMARY

[0003] Interpretation and execution of a customizable database request using an extensible computer process and an available computing environment is disclosed. In an aspect, a method includes generating an interpretation of a customizable database request which includes an extensible computer process and providing an input guidance to available processors of an available computing environment. The method further includes automatically distributing an execution of the interpretation across the available computing environment operating concurrently and in parallel, wherein a component of the execution is limited to at least a part of an input data. The method also includes automatically assembling a response using a distributed output of the execution.

[0004] The input guidance may be provided to each of the available processors and may be comprised of certain portions of the input data. The input guidance may be used to determine which of the available processors are to perform functions related to the at least the part of the input data. The method may further include providing an information to the extensible computer process about its context in the customizable database request, and processing an interpretation of the customizable database request based on the information provided. The extensible computer process may be a developer provided-computer program, and the information provided may include at least one of a format of the input data and an output data, whether the input data and the output data is ordered and in which form, grouping information, statistics of the input data and the output data, a distribution information, a length of the input data and the output data, and a custom parameter.

[0005] The custom parameter may be at least one of a number, a string, a list of numbers of strings, a content of a file in the available computing environment, and a result of the customizable database request. The method may further include post processing an output of each of the available processors when automatically assembling the response. The post processing may include at least one database operation including at least one of an aggregation operation, a sorting operation, and an invocation of another extensible computer process.

[0006] The method may further include pre-processing an input of each of the available processors when providing the input guidance to the available processors. The available computing environment may be comprised of at least two servers. The customizable database request may specify the input data for the extensible computer process. The input data may be structured in a form comprising at least one of a database table and an output of a different database query.

[0007] The input data may be unstructured in a form comprising a content of at least one file in a computing environment. The method may further include detecting a fault in the execution of the interpretation, and automatically rectifying an output effect of the fault. Rectifying the output effect of the fault may include at least one of reprocessing an operation, excluding a corrupted data, and logging the corrupted data. The customizable database request may be comprised of at least one of a predetermined function, a developer created function, and an analyst created function.

[0008] In another aspect, a system may include a query planning module to generate an interpretation of a database request which includes an extensible computer process, and a parallelization module to provide an information to available processors of an available computing environment and to automatically distribute an execution of the interpretation across the available computing environment operating concurrently and in parallel. A component of the execution may be limited to at least a part of an input data. The system may further include a response organization module to automatically assemble a response using a distributed output of the execution.

[0009] The information may be used to provide each of the available processors certain portions of the input data, and to determine which of the available processors are to perform functions related to the at least the part of the input data. The system may include a reference module to provide an extensible computer process information about its context in the database request. The system may include a dynamic interpretation module to process information that affects the interpretation of the database request based on the information provided, wherein the extensible computer process is a developer provided-computer program.

[0010] The information provided may include a format of the input data and an output data, whether the input data and the output data is ordered and in which form, grouping information, statistics of the input data, a distribution information, a length of the input data and the output data, and custom parameters. The custom parameters may be at least one of a number, a string, a list of numbers of strings, a content of a file in the available computing environment, and a result of the database request.

[0011] In yet another aspect, a method includes generating an interpretation of a customizable database request which includes an extensible computer process, and providing an input guidance to available processors of an available computing environment. The input guidance determines which of the available processors are to perform functions related to the at least a part of an input data. The method further includes pre-processing an input of each of the available processors when providing the input guidance to the available processors, and automatically distributing an analysis phase of the interpretation across the available computing environment operating concurrently and in parallel. A component of the analysis phase is limited to at least a part of the input data.

[0012] The method further includes automatically distributing an additional analysis phase of the interpretation across the available computing environment, and automatically assembling a response using a distributed output of the additional analysis phase. The method also includes post processing an output of each of the available processors when automatically assembling the response. The post processing includes at least one database operation including one or more of an aggregation operation, a sorting operation, and an invocation of another extensible computer process.

[0013] The method may include providing an extensible computer process information about its context in the customizable database request, and processing information that affects the interpretation of the customizable database request based on the information provided. In the aspect, the extensible computer process is a developer provided-computer program, and the information provided includes at least one of a format of the input data and an output data, whether the input data and the output data is ordered and in which form, grouping information, statistics of the input data, a distribution information, a length of the input data and the output data, and custom parameters. The custom parameters are one or more of a number, a string, a list of numbers of strings, a content of a file in the available computing environment, and a result of the customizable database request.

[0014] Other aspects and example embodiments are provided in the drawings and the detailed description that follows.

BRIEF DESCRIPTION OF THE VIEWS OF DRAWINGS

[0015] Example embodiments are illustrated by way of example and not limitation in the figures of the accompanying drawings, in which like references indicate similar elements and in which:

[0016] FIG. **1** is a system view illustrating processing of a customizable database query using a developer extensible operation and an available computing environment, according to one embodiment.

[0017] FIG. **2** is an exploded view of the available computing environment, according to one embodiment.

[0018] FIG. **3** is an exploded view of a query planning module, according to one embodiment.

[0019] FIG. **4** is an exploded view of a monitoring module, according to another embodiment.

[0020] FIG. **5** is an illustration of processing input data to generate a query response, according to another embodiment.

[0021] FIG. **6** is a system view of an alternate embodiment of processing of a customizable database query using a developer extensible operation and an available computing environment.

[0022] FIG. **7** is an illustration of processing input data to generate a query response, according to an alternate embodiment.

[0023] FIG. **8** is a diagrammatic system view of a data processing system in which any of the embodiments disclosed herein may be performed, according to one embodiment.

[0024] FIG. **9** is a process flow of interpreting and executing a customizable database request, according to one embodiment.

[0025] FIG. **10** is a process flow of automatically distributing an analysis phase and an additional analysis phase of the interpretation of a customizable database request across the available computing environment, according to one embodiment.

[0026] Other features of the present embodiments will be apparent from the accompanying drawings and from the detailed description that follows.

DETAILED DESCRIPTION

[0027] Interpretation and execution of a customizable database request using an extensible computer process and an available computing environment is disclosed. Although the present embodiments have been described with reference to specific example embodiments, it will be evident that various modifications and changes may be made to these embodiments without departing from the broader spirit and scope of the various embodiments.

[0028] FIG. **1** is a system view illustrating processing of a customizable database query using a developer extensible operation and an available computing environment, according to one embodiment. In particular, FIG. **1** illustrates an extensible computer process **100**, a query planning module **102**, an analysis phase **104**, an additional analysis phase **106**A-N, an available computing environment **112**, a monitoring module **114**, a response **116**, a user interface **118**, an analyst **120**, a developer **122**, a customizable database request **124**, and servers **126**.

[0029] FIG. **1** illustrates an analyst **120** providing a customizable database request **124** to a extensible computer process **100**. The analyst **120** may be a database analyst who is familiar with SQL (e.g., a Structured Query Language). SQL may be a database computer language designed for the retrieval and management of data in relational database management systems (RDBMS), database schema creation and modification, and database object access control management. The analyst **120** may have limited knowledge of other programming languages, and may have a substantially limited ability to create programs, to modify software, and to manage software distributed across multiple processors. The analyst **120** may be tasked with searching for data rather than developing programs.

[0030] The customizable database request **124** may consist of a SQL instruction and/or it may be written in any programming language. The customizable database request **124** may be customized to include a function (e.g., a nested SQL command, a mathematical equation, a variable, a standard deviation, etc.). The function may be created by the analyst **120**, the developer **122**, and/or it may be a predetermined function. The function may be customized to search multiple records at once, to retrieve and/or manipulate data in multiple forms (e.g., tables, images, unstructured data **584**, text files, programs, sound files, photos, etc.). The function may access data in one form and generate data in another form. The customizable database request **124** may further specify an input data **510** for the extensible computer process **100**.

[0031] The customizable database request **124** may allow the process to be scaled in accordance with a changing system hardware and/or performance of a system. The function may allow user-implemented procedural code to be uploaded to a database and executed at each node of a system. A user (e.g., an analyst **120**, a developer **122**, etc.) may provide code that may operate on individual rows and/or on groups of rows. The customizable database request **124** may take in input using a set of rows in a table (e.g., a persistent table in a database, the output of a SQL SELECT statement and/or the output of another function, etc.). The customizable database request **124** may result in an output that includes a relation of a set of rows (e.g., an output unrelated to the input.) The customizable database request **124** and/or a function of the customizable database request **124** may be placed into a SQL SELECT query and/or any other query as though it were itself a table. This integration with SQL may allow for composing SQL and procedural code invocations in any form and shape. The code may be written in Java, Python, and/or any other language.

[0032] In an embodiment, the customizable database request **124** may include a function that is written in Java that is then invoked as part of a SQL query statement. The function may convert sets of rows to sets of rows. The function may be parallelized to operate on rows across multiple nodes simultaneously. The function may be invoked on arbitrary sets of rows and/or rows grouped together by a PARTITION BY clause. Within a partition, rows may be further sorted using an ORDER BY clause.

[0033] In an embodiment, a function may split strings into words. In the embodiment, the function may be invoked once for every row in an input table. The function may include Java procedural code that takes each document and emits a row for each word. The function may define a column that appears in its output rows. In another embodiment, a function may be created to compute the 10 most-frequently occurring words in a body of text using the function to split strings into words.

[0034] In yet another embodiment, a function of the customizable database request **124** may perform sessionization by mapping each click in a clickstream to a unique session identifier. The function may define a session as a sequence of clicks by a particular user where no more than n seconds pass between successive clicks (e.g., if a click from a user isn't seen for n seconds, a new session is started. The function may use a userid and/or a timestamp attribute. The function may include as parameters the name of the timestamp attribute, the number of seconds between clicks that results in starting a new session. A clickstream table may be partitioned by userid, and partition tuples may be sequenced by timestamp. The sessionize function may then be invoked against each of the ordered partitions and/or emit the input tuples with an appropriate sessionid added.

[0035] The customizable database request **124** may be received by an extensible computer process **100**, which may be designed to take into consideration future growth by allowing the addition and/or modification of functionality. The addition of new functionality and/or the modification of existing functionality may be accomplished with limited impact to existing system functions. A developer **122** may be familiar with a type of programming involving database analysis, query modification, and/or data searches. The developer **122** may possess limited knowledge regarding programs to distribute an analysis across multiple computing systems. The developer **122** may support and/or design software for the analyst **120**. The developer **122** may adapt the extensible computer process **100** to add new functions, modify existing functions, and/or add additional language ability to the software.

[0036] The extensible computer process **100** may communicate with a query planning module **102** to generate a query interpretation of the customizable database request **124**. The query interpretation may be formatted to be distributable (e.g., separated into individual tasks for separate processes, etc.). The query interpretation may convert the customizable database request **124** from any computer language (e.g., a machine-readable artificial language designed to express computations that can be performed by a machine, C++, SQL, Perl, Java, Prolog, etc.) into a preferred programming language. The query interpretation may automatically format the customizable database query to be processed using a distributable, multiphase analysis.

[0037] The query planning module **102** may generate an interpretation (e.g., the query interpretation) of the customizable database request, which may include an extensible computer process. The query planning module **102** may optimize the analysis phase and/or the additional analysis phase using a parameter (e.g., an expected output file size, an input file format, a table dimension, etc.). The query planning module may provide an input guidance to available processors of the available computing environment. The input guidance may include certain portions of the input data, and the input guidance may be used to determine which of the available processors are to perform functions related to different parts of the input data.

[0038] The query planning module **102** may use the parameter to allocate a system resource (e.g., memory, power supply output, processor usage, a number of servers applied, a sequence of processors used, a timing of processes analyzed, etc.). The allocation of a system resource may include a distribution of processes across an available computing environment **112**, a selection of a type of analysis to apply, and/or a selection of input data to review. The execution of the interpretation may be automatically distributed across an available computing environment operating concurrently and in parallel, and a component of the execution may be limited to a part of the input data. The part of the input data may be a subset of the input data, which may allow the execution to be divided into separate tasks to be processed by different machines.

[0039] The available computing environment **112** (e.g., networked processors, virtual machines, multiple processors of a server, multiple servers **126**A-N and **128**A-N, etc.) may be comprised of servers that are and/or will be available to process data. The available computing environment **112** may be better illustrated in FIG. **2**.

[0040] The query interpretation may be dynamically determined based on a context (e.g., a repeated pattern of requested information, an association between an analyst's customizable database request **124** and an input data **510**, an available input data **510**, etc.). The context of the customizable database request **124** may include the type of requested information, the language of the request, and/or the expected response **116**. For example, if the analyst's request includes a name and address, the analysis phase **104** and/or the additional analysis phase **106**A-N may be adjusted to provide a response **116** that includes GPS coordinates (e.g., latitude and/or longitude, etc.). In another embodiment, the query interpretation may automatically provide alternate responses based on a variation of the requested parameters, such as by expanding or contracting a search parameter to provide alternate responses, varying search parameters, and searching for peak values.

[0041] The interpretation of the customizable database request generated by the query planning module **102** may be processed based on a contextual information provided to the extensible computer process. The extensible computer process may be a developer provided-computer program. The information provided may include a format of the input data and the output data, whether the input data and the output data are ordered and in which form, grouping information, statistics of the input data and the output data, a distribution information, a length of the input data and the output data, and a custom parameter.

[0042] The custom parameter may be a number, a string, and/or a list of numbers of strings. The custom parameter may further include a content of a file in the available computing environment, and/or a result of the customizable database request (e.g., the response **116**).

[0043] The query interpretation generated by the query planning module **102** may be communicated to an analysis phase **104**, which may be automatically distributed across an available computing environment **112**. The automatic distribution of the query interpretation may allow separate machines to analyze the query using portions of an input data **510** simultaneously, in parallel, in an overlapping sequence, and/or in series.

[0044] The analysis phase **104** may include a component that is limited to a part of the input data **510**. The component may process a part of a "map" phase of a MapReduce analysis (e.g., a framework for computing a distributable problem). The component may process a part of the analysis phase **104** using its part of the input data **510**. The analysis phase **104** may also include an additional component that uses the output of the component to generate an additional output (e.g., the additional component operates in series with the component, the additional component uses the output of the component as one of several inputs, etc.).

[0045] The analysis phase **104** may process the query interpretation using the input data **510**, which may be acquired from the database **108**A-N. The input data **510** may include structured data and/or unstructured data **584**, as illustrated in FIG. **5**. The input data of the analysis phase may be generated using a combination of multiple data sources (e.g., multiple tables, storage devices, etc.). The portion of the input data used by a component of the analysis phase **104** may also be generated using a combination of multiple data sources.

[0046] The analysis phase **104** may communicate with a monitoring module **114** and/or the additional analysis phase **106**A-N, which may be automatically distributed across the available computing environment (e.g., currently available servers, virtual machines, processors, etc.). The additional analysis phase **106**A-N may access a greater amount of information that the amount of the input data **510** used by the analysis phase **104**. The additional analysis phase **106**A-N may operate in parallel, in series, or in any other pattern with the analysis phase **104**.

[0047] The response **116** may be automatically assembled using a distributed output of the additional analysis phase **106**A-N. The output of the additional analysis phase **106**A-N may be distributed across multiple processors, servers, and/or virtual machines, and a complete resulting output may require an accumulation of all distributed parts of the additional analysis phase **106**A-N output. The assembled output may be the response **116**. The response **116** may be displayed through a user interface (e.g., a web browser, a terminal, a PC, a server, a monitor, etc.).

[0048] The monitoring module **114** may observe the input data **510** provided to the analysis phase **104**, the available computing environment **112**, the input to the additional analysis phase **106**A-N, the processing of information by the additional analysis phase **106**A-N, and the assembled response **116**. The monitoring module **114** may manage the automatic distribution of the analysis phase **104** and/or the additional analysis phase **106**A-N across the available computing environment **112**. The monitoring module **114** may assemble the distributed output of the additional analysis phase **106**A-N to generate the response **116**.

[0049] The monitoring module **114** may detect a fault (e.g., an exception, a hardware failure, a system crash, a processor failure, a data error, a processing error, etc.) in the analysis phase **104** and/or the additional analysis phase **106**A-N. The monitoring module **114** may automatically rectify an output effect (e.g., a data corruption, a propagating data error, a system failure, etc.) of the fault. The rectification may include one or more of reprocessing an operation (e.g., a component of the analysis phase **104**, the additional analysis phase **106**A-N, etc.), excluding a corrupted data, and/or logging a corrupted data. The rectification may include isolating a fault generating process and/or hardware mechanism. The monitoring module **114** may rectify an output effect automatically (e.g., without intervention by the developer **122** and/or analyst **120**).

[0050] FIG. **2** is an exploded view of the available computing environment **112** illustrated in FIG. **1**, according to one embodiment. In particular, FIG. **2** illustrates the available computing environment **112**, the servers **126**A-N, and the databases **108**A-N, according to one embodiment. The available computing environment **112** may include one or more servers that are currently or will be open to process information within a preferred time frame. The servers **126**A-N of the available computing environment **112** may be comprised of one or more separate servers, virtual machines, client devices, and/or separate processors of a single server. The servers **126**A-N may communicate with one or more databases (e.g., databases **108**A-N), which may be included within the available computing environment **112**. The servers **126**A-N and the databases **108**A-N may communicate with each other via a LAN, a WAN, a MAN, and/or any other network arrangement. In addition, the databases **108**A-N may include direct attached storage devices, volatile and/or non-volatile memory.

[0051] FIG. **3** is an exploded view of the query planning module, according to one embodiment. In particular, FIG. **3** includes the query planning module **102**, an optimization module **330**, a SQL instruction module **332**, a dynamic interpretation module **334**, a function module **336**, a developer operation module **338**, a translation module **340**, and a reference module **342**.

[0052] The query planning module **102** may include multiple modules to perform various functions. For example, the optimization module **330** may optimize the analysis phase **104** and/or the additional analysis phase **106**A-N using a parameter included with the customizable data request. The parameter may include a prediction and/or expectation regarding the response **116** (e.g., an output memory requirement, a number of generated responses, a range of response outputs, a type of input data **510**, etc.). The SQL instruction module **332** may interpret a SQL command, a nested SQL instruction, etc.

[0053] The dynamic interpretation module **334** may dynamically determine a query interpretation of the customizable database request **124** based on a context (e.g., a scope and/or format of the customizable database request **124**, an aspect of the input data **510**, the available computing environment, etc.). The analysis may be dynamically altered in accordance with the query interpretation.

[0054] The function module **336** may alter the query interpretation based on a function (e.g., a predetermined function, an analyst and/or developer created function, etc.). The function may be an equation, a programming command, a sequence of commands, etc. The developer operation module **338** may generate the query interpretation based on an operation added and/or modified by a developer in the extensible computer process **100**. The translation module **340** may generate the query interpretation by translating the customizable database request **124** from any language (e.g., a computer programming language such as SQL, Java, dBase, and/or a human language such as Indonesian, Russian, Spanish, and/or Chinese). The reference module **342** may provide an extensible computer process information about its context in the database request.

[0055] FIG. **4** is an exploded view of the monitoring module, according to another embodiment. In particular, FIG. **4** illustrates the monitoring module **114**, a detection module **450**, a rectification module **452**, a parallelization module **454**, an additional parallelization module **456**, and a response organization module **458**.

[0056] The detection module **450** may observe an input and/or an output of the analysis phase **104**, the servers **126**A-N, and the available computing environment **112**, the additional analysis phase **106**A-N. The detection module **450** may also observe the operation and transmitted data of the database **108**A-N, the query planning module, and/or the extensible computer process **100**. The detection module **450** may automatically detect a fault in the analysis phase **104** and/or the additional analysis phase **106**A-N.

[0057] The rectification module **452** may automatically rectify an output effect (e.g., a process failure, a system crash, a corrupted data, a propagating failure, etc.) of the fault. The automatic rectification may include an isolation of the fault generating mechanism (e.g., a process, a server, a component, etc.). The automatic rectification may include re-executing an interrupted process (e.g., the analysis phase **104**, the component, the additional analysis phase **106**A-N, etc.). The automatic rectification may include logging the fault and/or the corrupted data. The rectified data may be excluded (e.g., from a query response, a repeated analysis phase **104**, etc.).

[0058] The parallelization module **454** may automatically distribute the analysis phase of the query interpretation across an available computing environment. The additional parallelization module **456** may automatically distribute the additional analysis phase of the query interpretation across the available computing environment. The parallelization module **454** and/or the additional parallelization module **456** may consider a number of processors available, the number of analyses to be performed, and/or the sequence of the distributed processes.

[0059] The response organization module **458** may automatically assemble the response **116** using the distributed output of the additional analysis phase. The response organization module **458** may wait for a completion of all necessary processes prior to assembling the response **116**. The response organization module **458** may further post process an output of each of the available processors when automatically assembling the response. The post processing may include a database operation, such as an aggregation operation, a sorting operation, and/or an invocation of a separate extensible computer process (e.g., an external program, a developer created function, a third-party software, etc.).

[0060] FIG. **5** is an illustration of processing input data to generate a query response, according to another embodiment. In particular, FIG. **5** illustrates the analysis phase **104**, the additional analysis phase **106**, the input data **510**, the response **116**, a component **560**, an additional component **562**, a table **564**, text **566**, an object **568**, an audio file **570**, a video file **572**, an output table **574**, an output text **576**, an output object **578**, an audio file **580**, an output video file **582**, and an unstructured data **584**.

[0061] FIG. **5** illustrates a variety of types and forms that may be taken by the input data **510** and/or the response **116**. The input data **510** may include the table **564**, the text **566**, the object **568**, the audio file **570**, and/or the video file **572**. The input data may be structured in a form including a database table and/or an output of a different database query. The response **116** may include the output table **574**, the output text **576**, the output object **578**, the output audio file **580**, and/or the output video file **582**. The table **564** and/or the output table **574** may be structured data. The text **566**, the object **568**, the audio file **570**, the video file **572**, the output text **576**, the output object **578**, the output audio file **580**, and/or the output video file **582** may be unstructured data **584**. The input data **510** may be unstructured in a form including a content of at least one file in a computing environment. The unstructured data **584** may include a mix of data types, including images and audio files, text, programs, and/or word processing files.

[0062] The input data **510** may be communicated to the analysis phase **104**, which may process the data in the component **560** and/or the additional component **562**. The output of the analysis phase **104** may be received by the additional analysis phase **106**A-N, which may generate the response **116**. The additional analysis phase **106**A-N may consist of one or more phases. The response **116** may be formed of the same and/or a different data type from the input data **510**.

[0063] FIG. **6** is a system view of an alternate embodiment of processing of a customizable database query using a developer extensible operation and an available computing environment **112**. In particular, FIG. **6** illustrates the query planning module **102**, the analysis phase **104**, the additional analysis phase **106**, the database **108**, the input data **510**, the monitoring module **114**, the response **116**, the analyst **120**, the developer **122**, M **686**A-C, R **688**A-B, and intermediate files **690**.

[0064] The query planning module **102** may receive a customizable database request **124** from the analyst **120**. The developer **122** may contribute to and/or modify the customizable database request **124**. The query planning module **102** may communicate a query interpretation to the analysis phase **104**. The analysis phase **104** may receive an input data **510** from the database **108**. The input data **510** may be divided into split **0-4**. The analysis phase may include multiple components M **686**A-C. The additional analysis phase **106** may include the R **688**A-B. The M **686**A-C may each represent a map operation performed on a limited data input (e.g., split **0** and **1**, split **2** and **4**, split **3**, etc.). The M **686**A-C may generate intermediate files **690**, which may be communicated to the additional analysis phase **106**. The R **688**A-B may represent reduce operations in which the output of the map phases are accessible by each of the reduce operations. The R **688**A-B of the additional analysis phase **106** may produce output file **0-1** (e.g., the response **116**).

[0065] FIG. **7** is an illustration of processing input data to generate a response, according to an alternate embodiment. In particular, FIG. **7** illustrates the input data **510**, the analysis phase **104**, the additional analysis phase **106**, the response **116**, the developer **122**, the M **686**A-B and the R **688**A-B.

[0066] The input data **510** may include two text files (e.g., the dog, the cat). The analysis phase **104** may separate the text files into separate parts (e.g., the, dog, the, cat, etc.). The output of the operations M **686**A-B may be automatically redistributed to the parts of the additional analysis phase **106**. The outputs of the operations M **686**A-B may be sorted and/or categorized. The operations of the additional analysis phase, R **688**A-B may form the response **116**. The query response may include a count of each word (e.g., 1 "cat," 1 "dog," 2 "the," etc.). The M **686**A-B may each be limited to a part of the input data **510**. The R **688**A-B may be capable of accessing all outputs of the analysis phase **104**.

[0067] The developer **122** may customize and/or affect the operations (e.g., the M **686**A-B, the R **688**A-B, etc.) while the distribution of the analysis phase **104** and/or the additional analysis phase **106** are automatically handled.

[0068] FIG. **8** is a diagrammatic system view of a data processing system in which any of the embodiments disclosed herein may be performed, according to one embodiment. Particularly, the diagrammatic system view **800** of FIG. **8** illustrates a processor **802**, a main memory **804**, a static memory **806**, a bus **808**, a video display **810**, an alpha-numeric input device **812**, a cursor control device **814**, a drive unit **816**, a signal generation device **818**, a network interface device **820**, a machine readable medium **822**, instructions **824**, and a network **826**, according to one embodiment.

[0069] The diagrammatic system view **800** may indicate a personal computer and/or the data processing system in which one or more operations disclosed herein are performed. The processor **802** may be a microprocessor, a state machine, an application specific integrated circuit, a field programmable gate array, etc. (e.g., Intel® Pentium® processor). The main memory **804** may be a dynamic random access memory and/or a primary memory of a computer system.

[0070] The static memory **806** may be a hard drive, a flash drive, and/or other memory information associated with the data processing system. The bus **808** may be an interconnection between various circuits and/or structures of the data processing system. The video display **810** may provide graphical representation of information on the data processing system. The alpha-numeric input device **812** may be a keypad, a keyboard and/or any other input device of text (e.g., a special device to aid the physically handicapped).

[0071] The cursor control device **814** may be a pointing device such as a mouse. The drive unit **816** may be the hard drive, a storage system, and/or other longer term storage subsystem. The signal generation device **818** may be a bios and/or a functional operating system of the data processing system. The network interface device **820** may be a device that performs interface functions such as code conversion, protocol conversion and/or buffering required for communication to and from the network **826**. The machine readable medium **822** may provide instructions on which any of the methods disclosed herein may be performed. The instructions **824** may provide source code and/or data code to the processor **802** to enable any one or more operations disclosed herein.

[0072] FIG. **9** is a process flow of interpreting and executing a customizable database request, according to one embodiment. In operation **902**, an interpretation of a customizable database request may be generated (e.g., using the translation module **340** and/or the query planning module **102**), which may include an extensible computer process. In operation **904**, an input guidance may be provided to available processors of an available computing environment **112**. In operation **906**, an input of each of the available processors may be pre-processed (e.g., using the query planning module **102**) when providing the input guidance to the available processors. In operation **908**, an information may be provided to the extensible computer process about its context in the customizable database request (e.g., using the dynamic interpretation module **334** and/or the reference module **342**). In operation **910**, an interpretation of the customizable database request may be processed (e.g., using the query planning module **102**) based on the information provided. In operation **912**, an execution of the interpretation may be automatically distributed (e.g., using the analysis phase **104**) across the available computing environment operating concurrently and in parallel (e.g., using the reference module **342**). In operation **914**, a fault may be detected (e.g., using the detection module **450** of the monitoring module **114**) in the execution of the interpretation. In operation **918**, a response may be automatically assembled (e.g., by the response organization module **458**) using a distributed output of the execution. In operation **920**, an output of each of the available processors may be post processed (e.g., by the response organization module **458**) when automatically assembling the response.

[0073] FIG. **10** is a process flow of automatically distributing an analysis phase and an additional analysis phase of the interpretation of a customizable database request across the available computing environment, according to one embodiment. In operation **1002**, an interpretation of a customizable database request which includes an extensible computer process may be generated (e.g., using the SQL instruction module **332**, the translation module **340**, and/or the optimization module **330** of the query planning module **102**). In operation **1004**, an input guidance may be provided to available processors of an available computing environment. In operation **1006**, an extensible computer process information may be provided information about its context in the customizable database request (e.g., using the reference module **342**). In operation **1008**, an information may be processed (e.g., using the dynamic interpretation module **334**) that affects the interpretation of the customizable database request based on the information provided. In operation **1010**, an input of each of the available processors may be pre-processed when providing the input guidance to the available processors. In operation **1012**, an analysis phase of the interpretation may be automatically distributed (e.g., using the parallelization module **454**) across the available computing environment operating computing environment operating concurrently and in parallel. In operation **1014**, an additional analysis phase of the interpretation may be automatically distributed (e.g., using the additional parallelization module **456**) across the available computing environment. In operation **1016**, an output of each of the available processors may be post processed when the response is automatically assembled (e.g., using the response organization module **458**).

[0074] Although the present embodiments have been described with reference to specific example embodiments, it will be evident that various modifications and changes may be made to these embodiments without departing from the broader spirit and scope of the various embodiments. For example, the various devices, modules, analyzers, generators, etc. described herein may be enabled and operated using hardware circuitry (e.g., CMOS based logic circuitry), firmware, software and/or any combination of hardware, firmware, and/or software (e.g., embodied in a machine readable medium). For example, the various structures and methods may be embodied using transistors, logic gates, and electrical circuits (e.g., application specific integrated (ASIC) circuitry and/or in Digital Signal Processor (DSP) circuitry).

[0075] Particularly, the extensible computer process **100**, the query planning module **102**, the analysis phase **104**, the additional analysis phase **106**A-N, the monitoring module **114**, the user interface **118**, the optimization module **330**, the SQL instruction module **332**, the dynamic interpretation module **334**, the function module **336**, the developer operation module **338**, the translation module **340**, the reference module **342**, the detection module **450**, the rectification module **452**, the parallelization module **454**, the additional parallelization module **456**, the response organization module **458**, the component **560**, the additional component **562**, the M **686**A-C, and the R **688**A-B of FIGS. **1-10** may be enabled using software and/or circuitry.

[0076] In addition, it will be appreciated that the various operations, processes, and methods disclosed herein may be embodied in a machine-readable medium and/or a machine accessible medium compatible with a data processing system (e.g., a computer system), and may be performed in any order (e.g., including using means for achieving the various operations). Accordingly, the specification and drawings are to be regarded in an illustrative rather than a restrictive sense.

What is claimed is:

1. A method comprising:

generating an interpretation of a customizable database request which includes an extensible computer process;

providing an input guidance to available processors of an available computing environment;

automatically distributing an execution of the interpretation across the available computing environment operating concurrently and in parallel, wherein a component of the execution is limited to at least a part of an input data; and

automatically assembling a response using a distributed output of the execution.

2. The method of claim **1**, wherein the input guidance is provided to each of the available processors and is comprised of certain portions of the input data, and wherein the input guidance is used to determine which of the available processors are to perform functions related to the at least the part of the input data.

3. The method of claim **1** further comprising:

providing an information to the extensible computer process about its context in the customizable database request; and

processing an interpretation of the customizable database request based on the information provided, wherein the extensible computer process is a developer provided-computer program, and wherein the information provided includes at least one of a format of the input data and an output data, whether the input data and the output data is ordered and in which form, grouping information, statistics of the input data and the output data, a distribution information, a length of the input data and the output data, and a custom parameter.

4. The method of claim **3**, wherein the custom parameter is at least one of a number, a string, a list of numbers of strings, a content of a file in the available computing environment, and a result of the customizable database request.

5. The method of claim **1** further comprising post processing an output of each of the available processors when automatically assembling the response.

6. The method of claim **5** wherein the post processing includes at least one database operation including at least one of an aggregation operation, a sorting operation, and an invocation of another extensible computer process.

7. The method of claim **1** further comprising pre-processing an input of each of the available processors when providing the input guidance to the available processors.

8. The method of claim **1** wherein the available computing environment is comprised of at least two servers.

9. The method of claim **1**, wherein the customizable database request specifies the input data for the extensible computer process.

10. The method of claim **7**, wherein the input data is structured in a form comprising at least one of a database table and an output of a different database query.

11. The method of claim **7**, wherein the input data is unstructured in a form comprising a content of at least one file in a computing environment.

12. The method of claim **1**, further comprising:

detecting a fault in the execution of the interpretation; and

automatically rectifying an output effect of the fault.

13. The method of claim **12**, wherein rectifying the output effect of the fault includes at least one of reprocessing an operation, excluding a corrupted data, and logging the corrupted data.

14. The method of claim **1**, wherein the customizable database request is comprised of at least one of a predetermined function, a developer created function, and an analyst created function.

15. A system comprising:

a query planning module to generate an interpretation of a database request which includes an extensible computer process;

a parallelization module to provide an information to available processors of an available computing environment and to automatically distribute an execution of the interpretation across the available computing environment operating concurrently and in parallel, wherein a component of the execution is limited to at least a part of an input data; and

a response organization module to automatically assemble a response using a distributed output of the execution.

16. The system of claim **15**, wherein the information is used to provide each of the available processors certain portions of the input data, and to determine which of the available processors are to perform functions related to the at least the part of the input data.

17. The system of claim **15** further comprising:

a reference module to provide an extensible computer process information about its context in the database request; and a dynamic interpretation module to process information that affects the interpretation of the database request based on the information provided, wherein the extensible computer process is a developer provided-computer program, and wherein the information provided includes a format of the input data and an output data, whether the input data and the output data is ordered and in which form, grouping information, statistics of the input data, a distribution information, a length of the input data and the output data, and custom parameters.

18. The system of claim **17**, wherein the custom parameters are at least one of a number, a string, a list of numbers of strings, a content of a file in the available computing environment, and a result of the database request.

19. A method comprising:

generating an interpretation of a customizable database request which includes an extensible computer process;

providing an input guidance to available processors of an available computing environment, wherein the input guidance determines which of the available processors are to perform functions related to the at least a part of an input data;

pre-processing an input of each of the available processors when providing the input guidance to the available processors;

automatically distributing an analysis phase of the interpretation across the available computing environment operating concurrently and in parallel, wherein a component of the analysis phase is limited to at least a part of the input data;

automatically distributing an additional analysis phase of the interpretation across the available computing environment;

automatically assembling a response using a distributed output of the additional analysis phase; and

post processing an output of each of the available processors when automatically assembling the response, wherein the post processing includes at least one database operation including at least one of an aggregation operation, a sorting operation, and an invocation of another extensible computer process.

20. The method of claim **19** further comprising:

providing an extensible computer process information about its context in the customizable database request; and

processing information that affects the interpretation of the customizable database request based on the information provided, wherein the extensible computer process is a developer provided-computer program, and wherein the information provided includes at least one of a format of the input data and an output data, whether the input data and the output data is ordered and in which form, grouping information, statistics of the input data, a distribution information, a length of the input data and the output data, and custom parameters, wherein the custom parameters are at least one of a number, a string, a list of numbers of strings, a content of a file in the available computing environment, and a result of the customizable database request.

* * * * *