

(51) International Patent Classification:
G06F 17/30 (2006.01)(21) International Application Number:
PCT/EP2012/070819(22) International Filing Date:
19 October 2012 (19.10.2012)

(25) Filing Language: English

(26) Publication Language: English

(71) Applicant: TELEFONAKTIEBOLAGET L M ERIC-
SSON (PUBL) [SE/SE]; S-164 83 Stockholm (SE).(72) Inventors: PRASANNA KUMAR, Manoj; 6A,
Physiocare, Second Floor, Vinayaga Street, West Mambal-
am, Chennai 600033 (IN). SHIVASHANKAR, Sub-
ramanian; New No 4/12, Old No 3/189, Neithal Street,
Ponni Nagar, Karambakkam, Porur, Chennai 600116 (IN).
ZAHOOR, Jawad Mohamed; 1F, Salma Classic Terrace,
West Madha Church Street, Royapuram, Tamil Nadu,
Chennai 600013 (IN).(74) Agents: DAHNÉR, Christer et al.; Kransell & Wennborg,
P.O. Box 27834, S-115 93 Stockholm (SE).(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, IS, JP, KE, KG, KM, 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, 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,

[Continued on next page]

(54) Title: A FEDERATED DATABASE SYSTEM

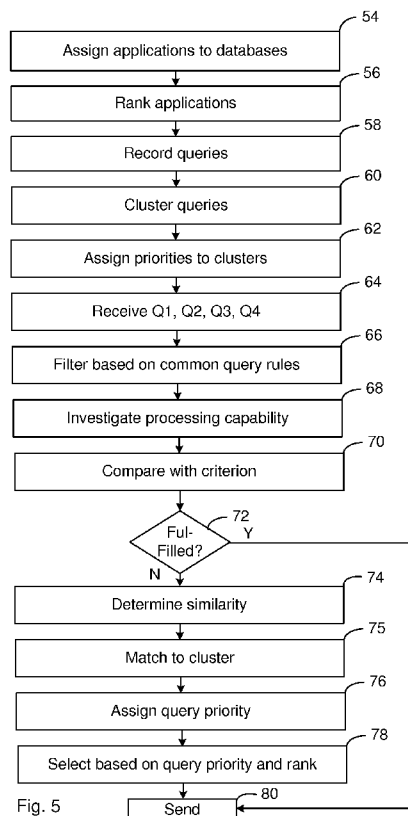


Fig. 5

(57) Abstract: The invention is directed towards a query handling device (20) in a federated database system (10) and a method, computer program and computer program product for handling queries in the federated database system. The query handling device (20) receives database queries (Q1, Q2, Q3, Q4) directed towards a group of databases (22, 24, 26, 28) from several applications (A1, A2, A3, A4) and filters the queries based on common query rules such as common query syntax, investigates the queries (Q1, Q2, Q3, Q4) that have passed the filtering, where the investigation comprises prioritising between at least some queries according to a query prioritisation scheme, and sends the investigated queries to the databases, where queries having been prioritised are sent according to the query prioritisation scheme.



TR), OAPI (BF, BJ, CF, CG, CI, CM, GA, GN, GQ, GW, ML, MR, NE, SN, TD, TG). **Published:**

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

Declarations under Rule 4.17:

— *as to applicant's entitlement to apply for and be granted a patent (Rule 4.17(ii))*

A FEDERATED DATABASE SYSTEM

TECHNICAL FIELD

5 The invention relates to federated database systems.
More particularly, the invention relates to a query
handling device in a federated database system as well
as to a method, computer program and computer program
product for handling queries in a federated database
10 system.

BACKGROUND

Systems like mobile telephone systems may comprise a
15 variety of databases of different types, which
databases may be queried by different applications,
such as by core applications in a core network or by
third party applications. This type of environment is
thus also a federated environment.

20

In recent times, software applications and users are
more concerned about efficiency of the databases with
regard to latency/delay. Users of these software
applications do not always prefer to explore the data
25 model of the database system to write optimized
queries. However, in order to write optimized queries,
users have to understand the capabilities and pitfalls
of the underlying data store of the database system
that the application uses.

30

Each such database has a data-store like MySQL, Oracle etc. in its data tier. Restrictions are imposed on the queries that are applied, depending on what data-store it uses in its data tier, as each data store has its own limitations and characteristics. For example, consider a database that uses MySQL as its data-store. It imposes query restrictions like "User cannot modify a table and select from the same table in a sub-query". Similarly, a system that uses Hbase as a data-store restricts joins in queries. Such query restrictions are imposed to ensure the effective usage of resources.

Apart from restrictions in constructing queries, other restrictions pertaining to data access are also imposed. For example some systems restrict a query from one user, say user A, to read/update data in a table created by another user, say user B. Such restrictions in queries protect privacy and integrity of the data.

In a federated database environment, multiple applications query multiple data-stores. In such an environment, it is hard to impose an exhaustive set of generic restrictions on queries that will optimize the performance in all data-stores.

Furthermore, in a federated database environment, there can be cases where an application might not have information about which data-store it is querying. In such cases, the problem of imposing query restrictions becomes more complex. More often a middleware takes care of data transfer from/to the application to/from

the underlying data-store. Still the middleware need not have knowledge about the data-store of the database that the application uses. Without knowing which data-store is used, it is difficult to impose query
5 restrictions.

Furthermore in such a federated system there is the risk of bottlenecks. Queries may have to wait unacceptable long times for their responses.

10

Furthermore, there is currently no way of handling queries in a federated system when there is a sudden outburst of queries.

15 There is thus a need for improvement with regard to some or all of the above-mentioned problems.

SUMMARY

20 One object of the invention is thus to provide an improvement in the handling of queries in a federated database system.

This object is according to a first aspect of the
25 invention achieved by a query handling device in a federated database system. The query handling device is configured to receive database queries directed towards a group of databases from several applications. It also comprises:

30 a query scheduler module configured to investigate the queries, where the investigation comprises prioritising

between at least some queries according to a query prioritisation scheme, and send the investigated queries to the databases. The prioritised queries are sent according to the query prioritisation scheme.

5

This object is according to a second aspect also achieved by a method for handling queries in a federated database system. The method is performed in a query handling device of the system and comprises the

10 steps of:

receiving database queries directed towards a group of databases from several applications,

investigating the queries, where the investigation comprises prioritising between at least some queries

15 according to a query prioritisation scheme, and

sending the investigated queries to the databases. The prioritised queries are sent according to the query prioritisation scheme.

20 The object is according to a third aspect of the invention also achieved through a computer program for handling queries in a federated database system. The computer program comprises computer program code which when run in a query handling device causes the query

25 handling device to:

receive database queries directed towards a group of databases from several applications,

investigate the queries, where the investigation comprises prioritising between at least some queries

30 according to a query prioritisation scheme, and

send the investigated queries to the databases. The prioritised queries are sent according to the query prioritisation scheme.

- 5 The object is according to a fourth aspect furthermore achieved by a computer program product for handling queries in a federated database system, where the computer program product is provided on a data carrier and comprises the computer program code according to
- 10 the third aspect.

The invention according to the above-mentioned aspects have a number of advantages. The prioritizing may remove bottlenecks. The use of prioritization may

15 remove starving conditions, where an end user might wait for a long time for the results of a query.

In an advantageous variation of the first aspect, the query scheduler module comprises a clustering unit that

20 clusters queries based on query execution time, assigns cluster priorities to clusters based on the execution times of the queries of the clusters, matches a query to a cluster and assigns a query priority to the query based on the cluster priority of the corresponding

25 cluster, a ranking unit that ranks the applications and a selection unit that select queries for execution based on query priority and application rank.

In a corresponding variation of the second aspect, the

30 method further comprises clustering queries based on query execution time, assigning cluster priorities to clusters based on the execution times of the queries of

the clusters and ranking the applications. In this variation the prioritising then comprises matching a query to a cluster, assigning a query priority to a query based on the cluster priority of the
5 corresponding cluster and selecting queries for execution based on query priority and application rank.

These variations provide the further advantage of providing a flexible association of a location with a
10 function in both time and space, which further enhances the flexibility with which locations can be selected and used.

According to a further variation of the first aspect,
15 the clustering unit determines the similarity of a received query with the clusters and matches the query to a cluster having similar queries.

According to a corresponding variation of the second
20 aspect, the prioritising further comprises determining the similarity of a received query with the clusters and matching the received query to a cluster having similar queries.

25 According to yet another variation of the first aspect, the ranking unit ranks an application based on the nature of the application, based on a database accessing frequency of the application and based on user priority.

30

According to a corresponding variation of the second aspect, the ranking of an application is made based on

the nature of the application, based on database accessing frequency of the application and based on user priority.

- 5 According to a further variation of the first aspect, the selecting unit prioritizes queries according to query priority and uses application rank if the query priorities of two or more queries are the same.
- 10 According to a corresponding variation of the second aspect, the prioritising of queries is made according to query priority. Application rank is then used if the query priorities of two or more queries are the same.
- 15 According to yet another variation of the first aspect, the query handling device further comprises a query filter module that filters the received queries based on common query rules, where the common query rules may comprise a common querying syntax.
- 20
- According to a corresponding variation of the second aspect, the method further comprises filtering the received queries based on common query rules, where the common query rules may comprise a common querying
- 25 syntax.

- According to a further variation of the first aspect, the query handling device also comprises a query handling evaluation module that investigates the
- 30 processing capability of system elements handling the query of an application, compares the processing capability with a capability criterion and informs the

query scheduler module of the result. The query scheduler module in turn allows the query to be immediately performed if the capability criterion is fulfilled and otherwise applies prioritisation
5 according to the query prioritisation scheme.

According to a corresponding variation of the second aspect, the investigating comprises investigating the processing capability of system elements handling a
10 query of an application, comparing the processing capability with a capability criterion. The query is then immediately performed if the capability criterion is fulfilled and otherwise the prioritisation is performed according to the query prioritisation scheme.
15

According to another variation of the first aspect, the query handling device comprises a reporting module that reports the progress of the queries to the applications.

20

It should be emphasized that the term "comprises/comprising" when used in this specification is taken to specify the presence of stated features, integers, steps or components, but does not preclude
25 the presence or addition of one or more other features, integers, steps, components or groups thereof.

BRIEF DESCRIPTION OF THE DRAWINGS

The invention will now be described in more detail in relation to the enclosed drawings, in which:

- 5 fig. 1 schematically shows a number of applications communicating with a number of databases via a query handling device,
- fig. 2 shows a block schematic of the query handling device,
- 10 fig. 3 shows a block schematic of a query scheduler module of the query handling device,
- fig. 4 shows a flow chart of a first embodiment of a method for handling queries in a federated database system,
- 15 fig. 5 shows a flow chart of a second embodiment of a method for handling queries in a federated database system, and
- fig. 6 schematically shows a computer program product comprising a data carrier with computer program code
- 20 for implementing the method for handling queries in a federated database system.

DETAILED DESCRIPTION

- 25 In the following description, for purposes of explanation and not limitation, specific details are set forth such as particular architectures, interfaces, techniques, etc. in order to provide a thorough understanding of the invention. However, it will be
- 30 apparent to those skilled in the art that the invention may be practiced in other embodiments that depart from these specific details. In other instances, detailed

descriptions of well-known devices, circuits and methods are omitted so as not to obscure the description of the invention with unnecessary detail.

5 The invention is generally directed towards a federated database system where there are several applications sending queries to several databases. In such a system an application should not have to know of the peculiarities of a certain database when providing
10 queries. It should thus be able to operate in the same way in respect of all databases irrespective of the different types that exist. Furthermore, queries should be handled in an efficient manner for avoiding bottlenecks.

15

In fig. 1 there is shown a federated database system 10, comprising a number of querying devices. There is here a first querying device 12, a second querying device 14, a third querying device 16 and a fourth
20 querying device 18. In each of these querying devices 12, 14, 16 and 18 there is furthermore running an application which sends queries destined for different databases. There is here a first application A1 in the first querying device 12, a second application A2 in
25 the second querying device 14, a third application A3 in the third querying device 16 and a fourth application A4 in the fourth querying device 18. The first application A1 sends a first query Q1, the second application A2 sends a second query Q2, the third
30 application Q3 sends a third query Q3 and the fourth application A4 sends a fourth query Q4. The federated database system 10 may furthermore be a part of a a

larger system, such as a mobile communication system, like Long Term Evolution (LTE) or Wideband Code Division Multiple Access (WCDMA). A mobile communication system may furthermore comprise a radio access network and a core network, where the core network may provide core network functionality such as keeping track of mobile terminals and their locations. This type of functionality may be based on one or more home location registers (HLR) that may need to use the databases of the federated database system 10. It is also known for various types of third party application servers connecting to such a mobile communication system. Also these servers may need to employ the federated database system 10.

15

The above mentioned queries are received by a query handling device 20, which in turn distributes them to various databases. For this reason the query handling device 20 communicates with a first database 22, with a second database 24, with a third database 26 and with a fourth database 28.

Fig. 2 shows a block schematic of the query handling device 20. It comprises an application interface 30, which is connected to a query filter module 32 as well as to a reporting module 38. The query filter module 32 is connected to a query handling evaluation module 34, which in turn is connected to a query scheduler module 36. Also the reporting module 38 is connected to the query scheduler module 36. The query scheduler module 46 and the reporting module 38 are further connected to

a database interface 40. There is finally an initiating module 31.

Here it may be mentioned that the applications A1, A2,
5 A3 and A4 may be provided in an application layer,
while the query handling device 20 and databases 22,
24, 26 and 28 may be provided in a federated data
layer.

10 Fig. 3 shows a block schematic of the query scheduler
module 36. It comprises a clustering unit 42 for
connection to the query handling evaluation module 34
as well as a ranking unit 44. These two units are
furthermore connected to a selection unit 46, which
15 selection unit 46 is to be connected to the database
interface as well as to the reporting unit.

A first embodiment will now be described with reference
being made to the previously mentioned figures as well
20 as to fig. 4, which shows a number of method steps in a
first embodiment of a method for handling queries in a
federated database system.

The query handling device 20 receives database queries
25 Q1, Q2, Q3, Q4 at the application interface 30, which
queries Q1, Q2, Q3, Q4 are directed towards the group
of databases 22, 24, 26, 28 from the different
applications A1, A2, A3, A4 in the querying devices 12,
14, 16 and 18, step 48. These queries should be handled
30 so that bottlenecks are avoided.

In order to enable this, the queries are provided to the query scheduler module 36, which investigates the queries Q1, Q2, Q3, Q4, step 50. This investigation may comprise prioritising between at least some queries
5 according to a query prioritisation scheme. The prioritisation scheme may use any of a number of different principles. It may for instance be based on the execution time of the query, where a short execution time may get a higher priority than a long
10 execution time. The execution time of a query may furthermore be predicted, and this prediction may be based on knowledge of known execution times of similar queries. The prioritization may furthermore be dynamic, i.e. only employed when there is a queue of queries in
15 the query handling device.

After the prioritisation the query scheduler module 36 sends the investigated queries to corresponding databases 22, 24, 26 and 28 via the database interface
20 40, step 52, where queries having been prioritised are thus sent according to the query prioritisation scheme.

It can in this first embodiment be seen that the reporting module 38, query filter module 32 and query
25 handling evaluation module 34 are not used, why these may be omitted from the query handling device.

Furthermore, the prioritizing may be made in order to avoid queries with lower execution time (high priority)
30 waiting for completion of queries with higher execution times (low priority). If there is no prioritization of this type, a simple query may take a longer time to

respond to due to the query being queued while a complex query is being completed. This might lead to starving conditions, where an end user might wait for a long time. Such a situation is thus avoided in the first embodiment.

Now a second embodiment will be described with reference being made to fig. 1 - 3 together with fig. 5, which shows a number of method steps in a method for handling queries in a federated database system according to the second embodiment.

The databases 22, 24, 26 and 28 all comprise processors such as central processing units (CPU) and data-stores in the form of memories.

Assume a federated system where applications do not know which data-store they query. In this situation the query handling device 20 takes care of data transfer from/to the application to/from a most suitable data-store through a dedicated connection.

In order to do this, the initiation module 31 assigns, at application deployment, the applications to data-stores of databases, step 54. The initiating module 31 may here choose the most optimum data-store for an application. The optimum data-store for the application may be chosen after analysing the data and queries of the application. For example, the initiating module 31 may choose HBase for an application with large volume of data and less relationship among entities requiring a low response time. The initiating module 31 then

- establishes a dedicated connection between the application and the chosen data-store. The query filter module 32 and query scheduler module 36 are then enabled for the connection. It may here be mentioned
- 5 that an assigned application is not aware of the type of database or data-store. The query handling device takes care of all the data transfer between the application and the data-store.
- 10 The ranking unit 44 of the query scheduler module 36 furthermore ranks the applications, step 56. The ranking is also typically performed at deployment of the applications.
- 15 The applications may be ranked based on initial sets of labels provided by a system administrator like $\text{Rank}(\text{Appplication1}) > \text{Rank}(\text{Application 2})$ and labels generated from application's behavioural pattern.
- 20 Applications may be ranked because their importance differ according to their complexity and usage. For example, a query from a core-network application may be more important than a query from a third party application for a telecom end-user of the federated
- 25 database architecture.

Basic information about the application is captured by the query handling device 20 when the application is deployed. Basic information may comprise nature of the

30 application (core application/third party application), database accessing frequency of the application such as read/write frequency. Basic ranking rules may thus be

predefined by the query handling device. For example, an application with high read/write frequency may receive a higher rank than an application with lower read/write frequency. Priority of users can also be
5 considered for ranking applications. For example, a user with a higher number of applications may get a higher priority than a user with a lower number of applications.

10 Thereafter the clustering unit 42 of the query scheduler module 36 is initialized. This is done through the query handling device 20 first recording queries from the different applications, step 58.

15 Queries are recorded for a period of time t defined by the query handling device 20. The execution time of the queries is also recorded.

Queries are then clustered by the clustering unit 42,
20 step 60. The clustering may be based on the execution time, i.e. queries with similar execution time may be grouped together.

The clustering may be based on the execution times of
25 the queries and performed using existing techniques like k-means clustering. According to these principles, each query is represented using the 1-d execution time and execution times are clustered using K-Means. This type of clustering is for instance described in the R
30 Journal, 2011, "Ckmeans.1d.dp: Optimal k-means Clustering in One Dimension by Dynamic Programming",

Haizhou Wang and Mingzhou Song, which is herein incorporated by reference.

Thereafter the clustering unit 42 assigns cluster
5 priorities to each of the resulting clusters based on execution time, step 62. Each cluster thus receives a corresponding cluster priority. The cluster priorities may furthermore be assigned to the resulting clusters based on the average execution time found for the
10 clusters. The cluster with queries having the shortest execution time may then get the highest cluster priority and the cluster with queries having the longest execution time may get the lowest cluster priority. Computed clusters may then be stored for
15 prioritizing queries in the future.

After these measures have been performed it is possible for the query handling device 20 to start to receive and handle queries in an efficient manner.

20

As in the first embodiment the method therefore then continues with the query handling device 20 receiving the queries Q1, Q2, Q3 and Q4 from the applications A1, A2, A3 and A4, step 64. The queries are also in this
25 embodiment received in the query interface 30. These queries are to be provided according to a general structure so that they can be handled for different database types. The queries should as an example be possible to use with MySQL, Oracle or HBase databases.

30

The queries Q1, Q2, Q3 and Q4 are then forwarded to the query filter module 32, which filters the queries based on common query rules, step 66.

5 The filtering may be a filtering where static restrictions are imposed on the queries. The query filter module 32 may for instance impose restrictions like "No regular expressions are allowed in keys". For example assume a query "Select name,age from my-
10 database where depart*name = 'Research'". This query may be filtered out by the query filter module 32 because it has a regular expression (depart*name) in the key 'department name'. The query filter module 32 sends the queries that have been filtered out to the
15 reporting module 38, which may inform the application sending the query that the query has been rejected.

Some of the possible restrictions are given below:

20 1. Multiple updates of a table in a single query is not allowed.
2. A common querying syntax should be followed as the system may communicate with any data-store according to the characteristics of the application. If a specific
25 querying syntax is followed, it might result in an error if the underlying data-store is compatible with a different syntax. For example, assume that an application generates a query "Select employeename from mydatabase where department = 'Research' ". This is a
30 valid SQL query. However, it might end up in an error if the underlying data-store is Hbase because Hbase

follows a different query syntax. The common query syntax can be as follows.

Select Query: Select tablename,
column1,column2...columnn

5 Insert Query: Insert tablename,
column1=value1,column2=value2

Delete Query: Delete tablename

3. No regular expressions are allowed in keys. For example, a query "Select employeename from my-database
10 where depa*='Research' " is not allowed because it has a regular expression (depa*) in a row key department.

4. Reading multiple tables in a single query is not permitted as it is not allowed in certain data-stores like Hbase.

15 5. Kindless queries that try to view all the entities in a data-store are not allowed as data created by other applications and the system (query handling device) should not be visible to the application.

6. Filters on more than one property is not allowed as
20 certain data-stores like Hbase do not capture relationships among properties.

Queries following these restrictions are accepted and forwarded to the query handling evaluation unit 34.

25

The query handling evaluation unit 42 then investigates the processing capability of the system elements handling the query for an application, step 68.

30 The assigned data-stores have system elements like processors, such as CPUs and memories that are to handle the queries. The query handling evaluation unit

42 investigates the processing capability of these system elements. It does this through monitoring various CPU and memory parameters of these system elements. If the database to which the query is addressed is idle, then the query is scheduled for execution immediately. However, if the system is busy, the query is prioritized by the query scheduler module 36.

10 The investigating of processing capability may comprise investigating various Central Processing Unit (CPU) and memory parameters of the data-store in question. The CPU parameters monitored may be CPU Load average, available free RAM space and cache size. The memory parameter monitored may be the available space in the disk. If for instance the first application A1 has been assigned to the first database 22, the processor and memory parameters for the data-store of this database are investigated.

20

The processing capability for the queries are then compared with a processing capability criterion, step 70, and if the criterion is fulfilled, step 72, the query is immediately sent to the assigned database without being handled by the query scheduler module 36, step 80. The criterion may for instance be considered to be fulfilled if the system that contains the data-store of the application is free i.e., the CPU load average is less than a first threshold T1 and the available free RAM space is greater than a second threshold T2, where the threshold values T1 and T2 may

be set by the query handling device 20 based on the type of processor that the data-store uses.

If the criterion is not fulfilled, step 72, i.e. the system is considered to be busy, then the query scheduler module 36 is informed and the queries are forwarded to this module.

In the query scheduler module 36, queries are immediately forwarded to the database interface 40, if the corresponding criterion is fulfilled, step 62. However, if the criterion is not fulfilled, step 62, the clustering unit 42 of the query scheduler module 36 handles the queries.

15

The clustering unit 42 first matches a query to a cluster, step 74, which may be done via a determination of the similarity of a query to the different clusters.

20 The Execution time of the query may be predicted by computing the similarity between the query and all other queries near the centroid of the clusters.

- Similarity may be computed by using existing techniques like cosine similarity, for instance as described in Journal of Machine Learning Research 10 (2009) 747-776, "Similarity-based Classification: Concepts and Algorithms", Yihua Chen, Eric K. Garcia, Maya R. Gupta, Ali Rahimi and Luca Cazzanti, which is herein incorporated by reference.

30

Attributes like fields queried and type of query may be considered for computing similarity.

After the similarity has been determined, the query may thus be matched to the cluster that is most similar, step 75. The query may then be assigned a query
 5 priority that is based on the cluster priority of the corresponding cluster, step 76, i.e. that is based on the cluster priority of the cluster that in the matching was found to be most similar to the investigated query. This may be done through assigning
 10 the query to the cluster. Thereby the query priority given to the query may be the same as the cluster priority of the associated cluster, which associated cluster may thus be a cluster having queries that are similar to the query in question.

15

A query may thus be assigned to a cluster that has similar queries. Assume that $(x_1, x_2, \dots, x_n)$ is a set of execution times of queries $(q_1, q_2, \dots, q_n)$. K-means clustering algorithm partitions n observations of
 20 execution times into k clusters. Given a set of k centroids $m_1^{(1)}, \dots, m_k^{(1)}$ and a query x_p , the algorithm does the following computation.

$$S_i^{(t)} = (x_p : \|x_p - m_i^{(t)}\| \leq \|x_p - m_j^{(t)}\| \forall 1 \leq j \leq k) \quad (1)$$

25

where each x_p is assigned to only one cluster $s_i(t)$.
 - Once query is assigned to a cluster, its execution time is predicted by

$$30 \quad E(Q_i) = \frac{E(Q_1) + E(Q_2) + \dots + E(Q_n)}{n} \quad (2)$$

Note that this is the centroid value of the corresponding cluster which is already pre-computed.

5 where,

$E(Q_t)$ = Execution time of a query Q_t

n = Number of queries in cluster

and the priority of the query will be the same as the
10 priority of the cluster.

It can in this way be seen that the clustering unit 42 may assign a query priority to a query that is the same as the cluster priority of a cluster having similar
15 queries.

After prioritization of queries, they are selected for execution by the selecting unit 46 according to their query priority while considering the rank of the
20 corresponding application, step 78. This means that the selecting unit 46 connected to the ranking unit 44 in order to obtain the ranks of the applications for being used when selecting queries.

25 In this way queries are scheduled based on the ranks of the applications from which they originate and their query priorities.

A query is thus scheduled for execution based on the
30 query priority. If a query with high query priority is waiting in queue for a query with a lower query priority to complete execution, the query with lower

query priority can be preempted if the underlying data-store supports preemption. Conflicts in priorities of queries may be resolved by considering the rank of the application that generated the queries. For example,

5 assume query Q1 and query Q2 have rank 1. Q1 is generated by application A1 and Q2 is generated by application A2. Also assume $R(A1) > R(A2)$ where R is the rank of an application. In this case, Q1 is executed first because the application A1 that

10 generated the query has a higher rank. The selecting unit thus prioritizes queries according to query priority and uses application rank if the query priorities of two or more queries are the same. In case there is a common conflict in the priorities of queries

15 and rank of applications, a query is selected at random and scheduled for execution. Assume that $R(A1) = R(A2)$ in the above example. In this case Q2 might be selected at random for execution.

20 After a query has been selected in this way it is then sent to the associated database, step 80, and the results will appear in the data-store that is associated with the application.

25 Furthermore, while executing an insert query, the available space in the disk associated with the query may be considered. If the system containing the data-store of the application does not have sufficient space to store the records to be inserted, then the query may

30 be buffered until the disk capacity is increased.

Throughout the operation of the query handling device 20, the reporting module 38 monitors and reports the status of the queries to the applications A1, A2, A3 and A4. It does for instance report the rejected
5 queries and also the approximate time of execution of accepted queries.

The reporting module 38 thus informs the applications about the status of queries. Messages used by the
10 reporting module may be as follows.

1. Query Executed.
2. Query Rejected due to violation in restrictions.
3. Query scheduled for execution shortly

15

Based on the status of the query, one among the above mentioned messages may be displayed to inform the application about the status of the query.

20 It may be seen that the query handling device improves the performance of a federated database system by imposing static and dynamic query restrictions.

The invention also has a number of further advantages.
25 Through employing static restrictions on queries, the query handling device ensures that queries generated by the applications don't result in an error due to syntax issues. Through employing dynamic restrictions on queries a performance bottleneck in the part of the
30 system where the data-store is provided is avoided. Query level scheduling ensures high performance and stability as system failure due to overload is avoided

completely. Through combining static and dynamic restrictions both syntactic conflicts and performance issues are resolved in the federated database architecture. The use of application ranking ensures
5 that critical applications are always stable in terms of data update/retrieval. The use of query prioritizing with support for preemption avoids queries with lower execution times (high priority) waiting for completion of queries with longer execution times (low priority).
10 If there is no query prioritizing, a scenario may occur where the responding time for a simple query is excessively long due to the waiting time for completion of the complex query. This might lead to starving conditions, where an end user might wait for a long
15 time. The proposed method can avoid such scenarios. Furthermore, the stability of the application and the system is guaranteed in case of sudden outburst of queries. For example, a Home Location Register (HLR) may perform a large number of batch inserts during
20 night time.

The query handling device may be provided in the form of a processor with associated program memory including computer program code for performing the functionality
25 of the query handling device.

A computer program may also be a computer program product, for instance in the form of a computer readable storage medium or data carrier, like a CD ROM
30 disc or a memory stick, carrying such a computer program with the computer program code, which will implement the functionality of the above-described

query handling device. One such data carrier 82 with computer program code 84 is schematically shown in fig. 6.

5 While the invention has been described in connection with what is presently considered to be most practical and preferred embodiments, it is to be understood that the invention is not to be limited to the disclosed
embodiments, but on the contrary, is intended to cover
10 various modifications and equivalent arrangements.
Therefore the invention is only to be limited by the following claims.

CLAIMS

1. A query handling device (20) in a federated database system (10), the query handling device
5 being configured to receive database queries (Q1, Q2, Q3, Q4) directed towards a group of databases (22, 24, 26, 28) from several applications (A1, A2, A3, A4) and comprising:
a query scheduler module (36) configured to
10 investigate the queries (Q1, Q2, Q3, Q4), where the investigation comprises prioritising between at least some queries according to a query prioritisation scheme, and send the investigated queries to the databases, where queries having been
15 prioritised are sent according to the query prioritisation scheme.
2. The query handling device according to claim 1, wherein the query scheduler module (36) comprises a
20 clustering unit (42) configured to cluster queries based on query execution time, assign cluster priorities to clusters based on the execution times of the queries of the clusters, match a query to a cluster and assign a query priority to the query based
25 on the cluster priority of the corresponding cluster, a ranking unit (44) configured to rank the applications and a selection unit (46) configured to select queries for execution based on query priority and application rank.
- 30 3. The query handling device according to claim 2, wherein the clustering unit (42) is configured to

determine the similarity of a received query with the clusters and match the received query to a cluster having similar queries.

5 4. The query handling device according to claim 2 or 3, wherein the ranking unit (44) is configured to rank an application based on the nature of the application, based on a database accessing frequency of the application and based on user priority.

10 5. The query handling device according to any of claims 2 - 4, wherein the selecting unit (46) is configured to prioritize queries according to query priority and to use application rank if the query
15 priorities of two or more queries are the same.

20 6. The query handling device according to any previous claim, further comprising a query filter module (32) configured to filter the received
20 queries based on common query rules.

25 7. The query handling device according to claim 6, wherein the common query rules comprises a common querying syntax.

30 8. The query handling device according to any previous claim, further comprising a query handling evaluation module (34) configured to investigate the processing capability of system elements handling
30 the query of an application, compare the processing capability with a capability criterion and inform the query scheduler module (36) of the result, where

the query scheduler module (36) is configured to allow the query to be immediately performed if the capability criterion is fulfilled and otherwise to apply prioritisation according to the query
5 prioritisation scheme.

9. The query handling device according to any previous claim, further comprising a reporting module (38) configured to report the progress of the
10 queries to the application.

10. A method for handling queries in a federated database system (10), the method being performed in a query handling device (20) of the system and
15 comprising the steps of:
receiving (48; 64) database queries (Q1, Q2, Q3, Q4) directed towards a group of databases (22, 24, 26, 28) from several applications (A1, A2, A3, A4),
investigating (50; 66, 68, 70, 72, 74, 75, 76) the
20 queries (Q1, Q2, Q3, Q4), where the investigation comprises prioritising between at least some queries according to a query prioritisation scheme, and
sending (52; 80) the investigated queries to the databases, where queries having been prioritised are
25 sent according to the query prioritisation scheme.

11. The method according to claim 10, further comprising clustering (60) queries based on query execution time, assigning (62) cluster priorities to
30 clusters based on the execution times of the queries of the clusters, ranking (56) the applications and the prioritising comprises matching (75) a query to

a cluster, assigning (76) a query priority to the query based on the cluster priority of the corresponding cluster and selecting (78) queries for execution based on query priority and application rank.

12. The method according to claim 11, wherein the prioritising further comprises determining (74) the similarity of a received query with the clusters and the matching comprises matching the query to a cluster having similar queries.

13. The method according to claim 11 or 12, wherein the ranking of an application is made based on the nature of the application, based on database accessing frequency of the application and based on user priority.

14. The method according to any of claims 11 - 14, wherein the prioritising of queries is made according to query priority and application rank is used if the query priorities of two or more queries are the same.

15. The method according to any of claims 10 - 14, further comprising filtering (56) the received queries based on common query rules.

16. The method according to any of claims 10 - 15, wherein the investigating comprises investigating (68) the processing capability of system elements handling a query of an application, comparing (70)

the processing capability with a capability criterion, where the query is immediately performed if (72) the capability criterion is fulfilled and otherwise (72) the prioritisation is performed
5 according to the query prioritisation scheme.

17. A computer program for handling queries in a federated database system (10) and comprising computer program code (84) which when run in a query
10 handling device (20), causes the query handling device to:

receive database queries (Q1, Q2, Q3, Q4) directed towards a group of databases (22, 24, 26, 28) from several applications (A1, A2, A3, A4),
15 investigate the queries, where the investigation comprises prioritising between at least some queries according to a query prioritisation scheme, and send the investigated queries to the databases, where queries having been prioritised are sent
20 according to the query prioritisation scheme.

18. A computer program product for handling queries in a federated database system (10), said computer program product being provided on a data carrier
25 (82) and comprising said computer program code (82) according to claim 17.

1/4

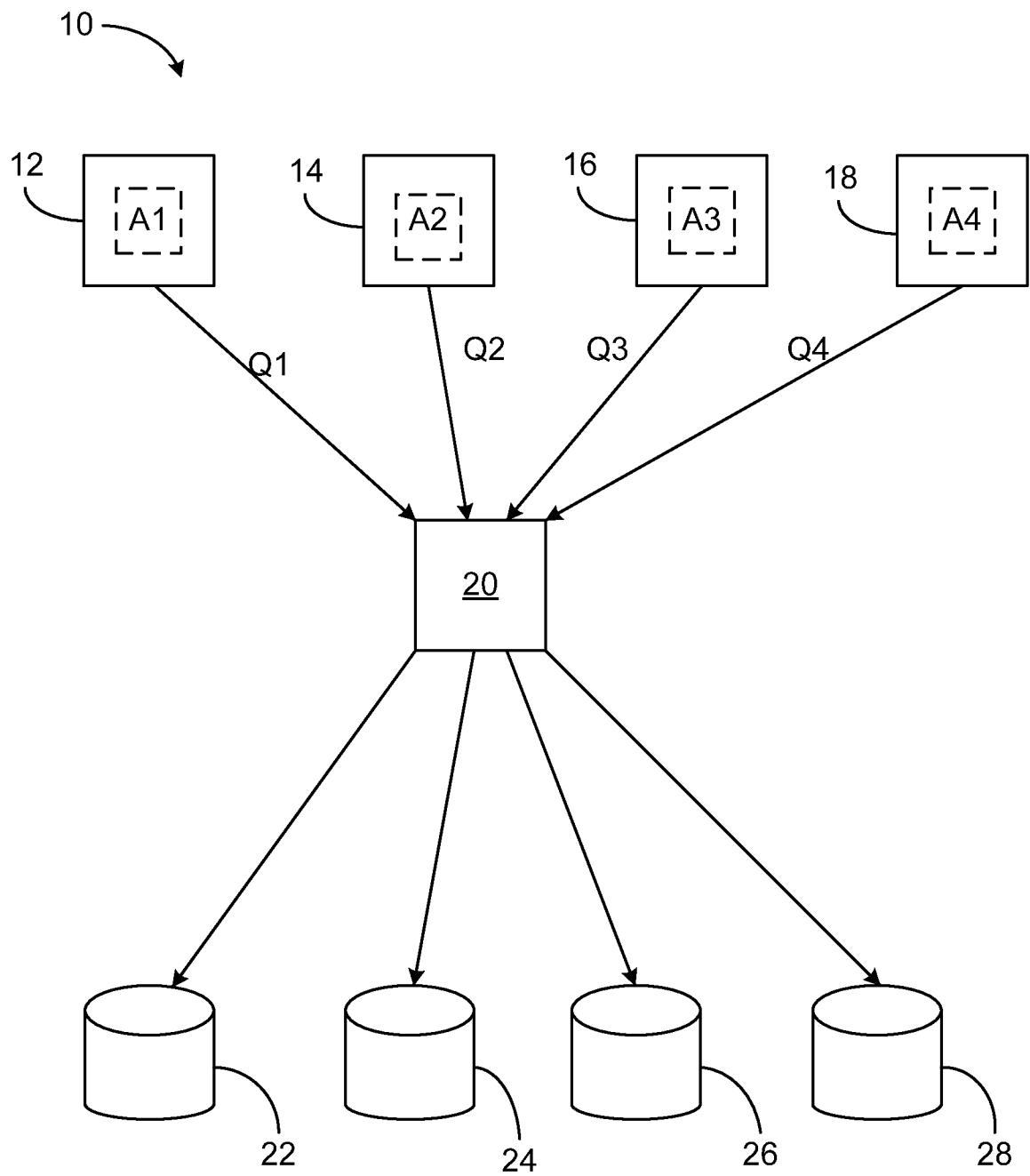


Fig. 1

2/4

20
↘

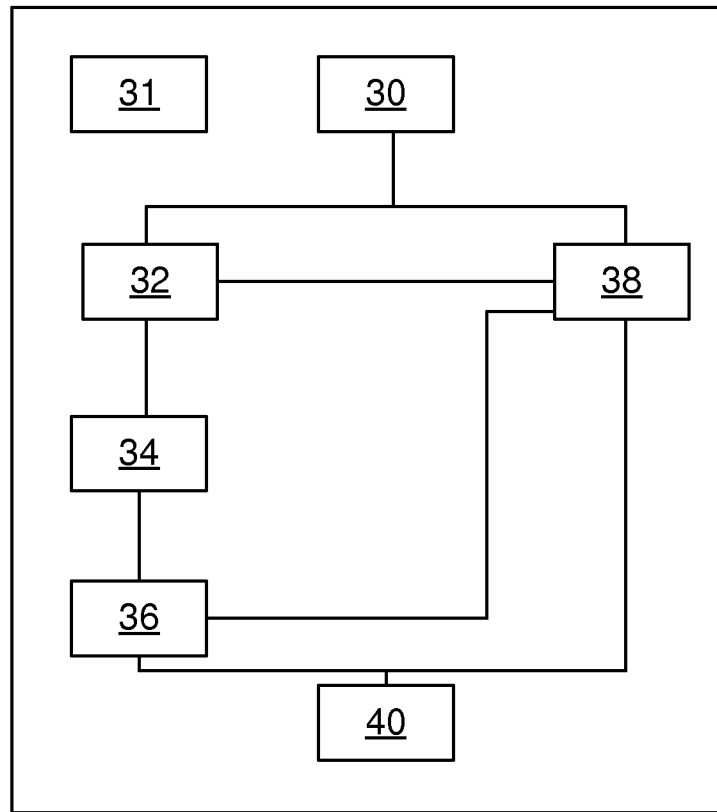


Fig. 2

36
↘

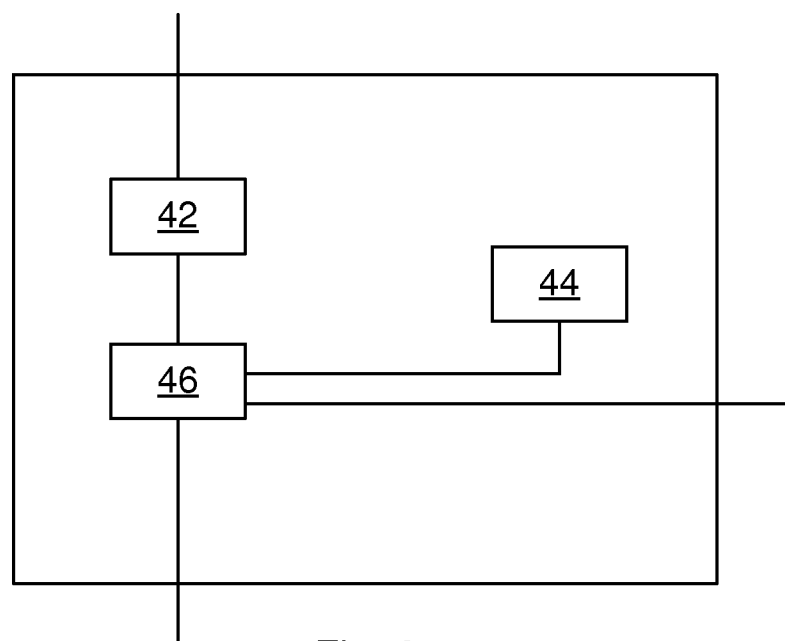


Fig. 3

3/4

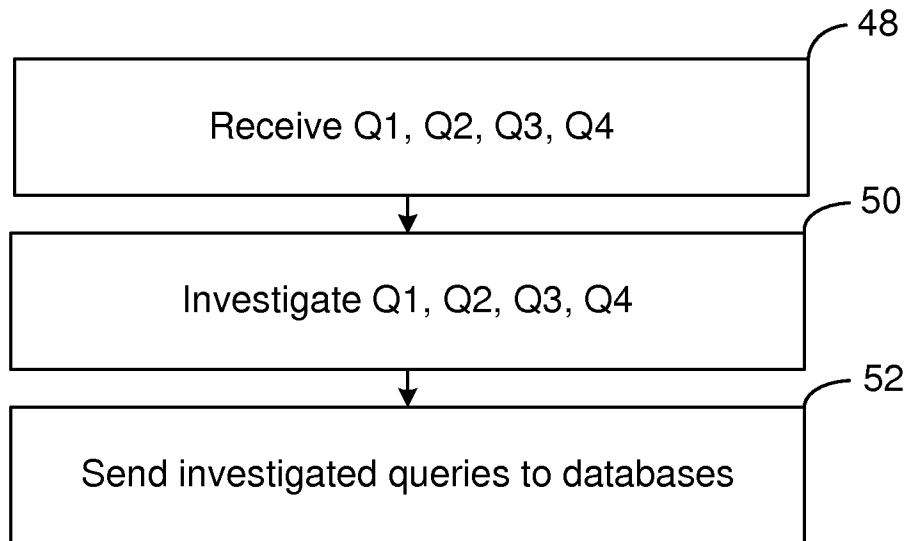


Fig. 4

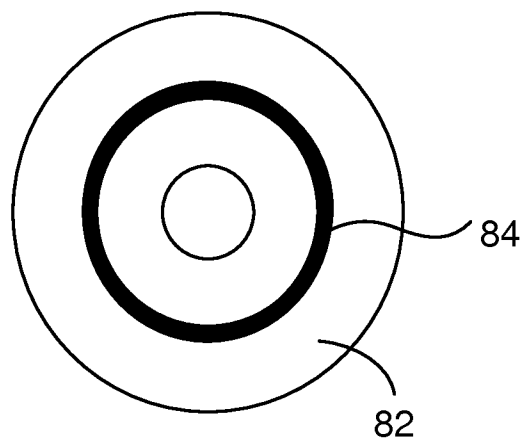


Fig. 6

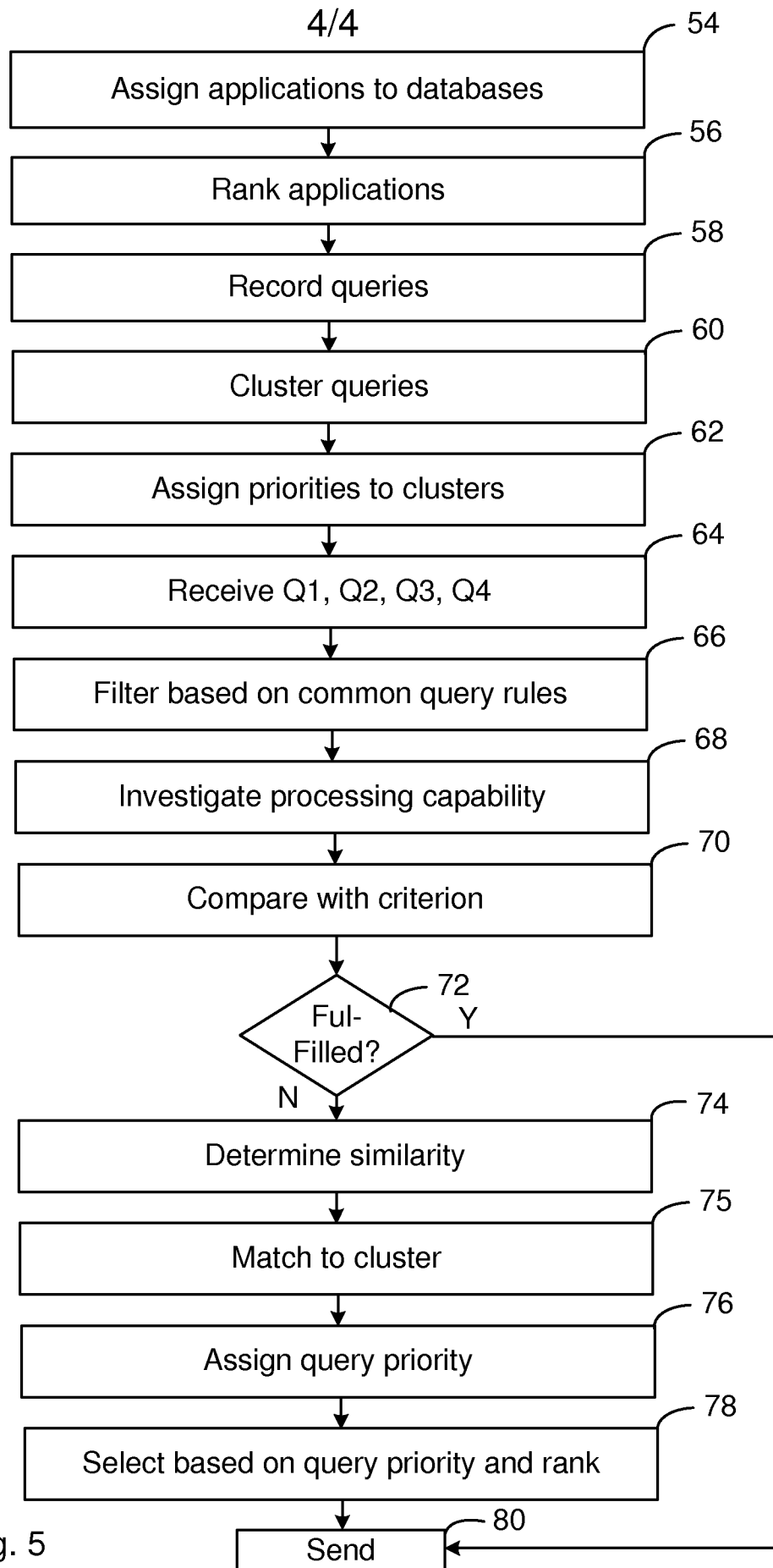


Fig. 5

INTERNATIONAL SEARCH REPORT

International application No

PCT/EP2012/070819

A. CLASSIFICATION OF SUBJECT MATTER

INV. G06F17/30

ADD.

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

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

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

EPO-Internal

C. DOCUMENTS CONSIDERED TO BE RELEVANT

Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
X	US 2004/186829 A1 (SUZUKI YOSHIO [JP] ET AL) 23 September 2004 (2004-09-23) abstract paragraphs [0005], [0020] - [0026], [0053], [0058] - [0062], [0070] - [0076] -----	1-18
X	US 2009/132611 A1 (BROWN DOUGLAS [US] ET AL) 21 May 2009 (2009-05-21) abstract paragraphs [0033] - [0035], [0046] - [0090] -----	1-18



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

27 June 2013

Date of mailing of the international search report

05/07/2013

Name and mailing address of the ISA/

European Patent Office, P.B. 5818 Patentlaan 2
NL - 2280 HV Rijswijk
Tel. (+31-70) 340-2040,
Fax: (+31-70) 340-3016

Authorized officer

Hackelbusch, Richard

INTERNATIONAL SEARCH REPORT

Information on patent family members

International application No

PCT/EP2012/070819

Patent document cited in search report	Publication date	Patent family member(s)	Publication date
US 2004186829 A1	23-09-2004	JP 4327481 B2	09-09-2009
		JP 2004280528 A	07-10-2004
		US 2004186829 A1	23-09-2004

US 2009132611 A1	21-05-2009	NONE	
