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

(19) World Intellectual Property Organization
International Bureau

(43) International Publication Date
6 August 2009 (06.08.2009)

PCT

(10) International Publication Number
WO 2009/095083 A1

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

(21) International Application Number:
PCT/EP2008/051202

(22) International Filing Date: 31 January 2008 (31.01.2008)

(25) Filing Language: English

(26) Publication Language: English

(71) Applicant *(for all designated States except US)*: **TELEFONAKTIEBOLAGET LM ERICSSON (publ)** [SE/SE]; S-164 83 Stockholm (SE).

(72) Inventors; and

(75) Inventors/Applicants *(for US only)*: **LARSSON, Tony** [SE/SE]; Nils Åhlins väg2, S-194 76 Upplands Väsby (SE). **LIDSTRÖM, Mattias** [SE/SE]; Kungsholms Kyrkoplan 3A, S-112 24 Stockholm (SE). **MATTI, Mona** [SE/SE]; Danielsvägen 2A, S-131 40 Nacka (SE). **SVENSSON, Martin** [SE/SE]; Vingårdsgatan 7, S-117 59 Stockholm (SE).

(74) Agent: **KARLSSON, Leif**; Ström & Gulliksson AB, P.O Box. 4188, S-203 13 Malmö (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, BR, BW, BY, BZ, CA, CH, 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, PG, PH, PL, PT, RO, RS, RU, SC, SD, SE, SG, SK, SL, SM, SV, SY, 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, LS, MW, MZ, NA, SD, SL, SZ, TZ, UG, ZM, ZW), Eurasian (AM, AZ, BY, KG, KZ, MD, RU, TJ, TM),

[Continued on next page]

(54) Title: LOSSY COMPRESSION OF DATA

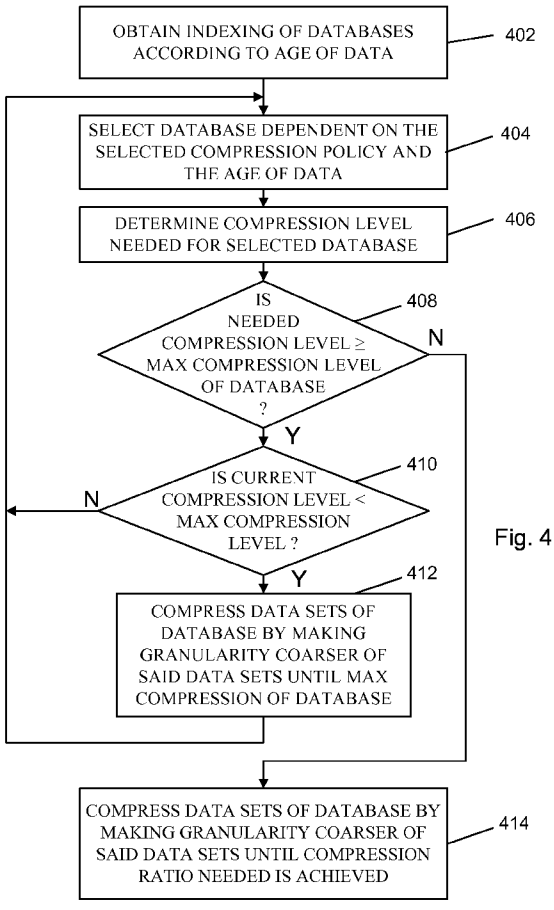

Fig. 4

(57) Abstract: The present invention provides a method and a data processing arrangement (100) for increasing the data storage efficiency of at least a first database of a data repository (114) comprising at least two databases. Based on information at least related to the age of the data of the databases, at least a first database is selected (step 404), after which at least a first data set of at least the first database is compressed (steps 412, 414) by making the granularity coarser of at least the first data set, such that the data storage efficiency in the data repository is increased. This brings the advantage that a data storage capacity is used more efficiently and that data is quickly and easily accessible without requiring unpacking of data.

WO 2009/095083 A1

European (AT, BE, BG, CH, CY, CZ, DE, DK, EE, ES, FI, FR, GB, GR, HR, HU, IE, IS, IT, LT, LU, LV, MC, MT, NL, NO, PL, PT, RO, SE, SI, SK, TR), OAPI (BF, BJ, CF, CG, CI, CM, GA, GN, GQ, GW, ML, MR, NE, SN, TD, TG).

Declarations under Rule 4.17:

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

— *of inventorship (Rule 4.17(iv))*

Published:

— *with international search report*

LOSSY COMPRESSION OF DATA

TECHNICAL FIELD

The present invention relates in general to compression of data and in particular to a method and an arrangement for lossy compression of log data, providing a more efficient usage of a data storage capacity.

BACKGROUND

Communication companies, such as telephone operators or Internet Service Providers (ISPs) often generate huge amounts of log data about how networks are being used, in order to attempt improving their service towards customers. These log data are typically stored in databases for a period of time, after which the data finally are removed from the databases, simply for the reason that the storage capacity in not sufficiently high.

In order to use storage capacity of databases more efficiently, several approaches can be used. First of all, data may be compressed, for instance, by using standard algorithms such as zip. Secondly, log data may be analyzed for finding redundant information or for removing information that is irrelevant for the type of analysis that is to be conducted. Thirdly, all data may be deleted except for samples of log data that are kept for future analysis.

There are however a few problems with current methods of using storage capacity more efficiently. One problem relates to the fact that compressing and uncompressing of data are time consuming activities, which often bring the problem of waiting times and delays to a user using such techniques. Another problem is that data in general are compressed in batches or large data sets, and that it is usually not possible to extract data from the batch without having to uncompress the entire batch or large data set, which indeed can be time consuming. That is, it is difficult to provide indexed retrieval of information. In the case information is removed from a database, certainly being one

way to reduce the size of database, future analysis requiring the removed information cannot be carried out completely or may even be entirely disabled.

There is therefore a need for a method and an arrangement for using data storage capacity more efficiently, which circumvent or at least diminish the problems as mentioned above.

SUMMARY

An object of the present invention is to provide an efficient method for compression of data of a database.

This object is solved by a method for increasing the data storage efficiency of at least a first database of a data repository comprising at least two databases. This method comprises the step of obtaining information at least related to the age of the databases, the step of selecting at least a first database in dependence of the age of the databases, and the step of compressing at least a first data set of at least the first database by making the granularity coarser of at least the first data set.

This method has the advantage of enabling an indexed retrieval of stored data, without the need to un-pack the compressed data to gain access to said data.

The step of selecting at least a first database, within the method for increasing the data storage efficiency, may further comprises selecting at least the first database according to decreasing age of the databases.

It is further an advantage to compress data dependent of the age of the data, such that new data, which often are regarded as more important than old data, may be given a higher relevance.

The step of obtaining information, within the method for increasing the data storage efficiency, may further comprises obtaining information related to a compression policy, the step of selecting may further comprise selecting the first database in dependence of the compressing policy, and the step of compressing may further comprise applying the compression policy for compressing at least the first data set.

Making the granularity coarser within the step of compressing at least a first data set, in the method for increasing the data storage efficiency, may comprise converting at least the first data set into one or more representations of said data set in one or more data dependent intervals.

The step of selecting in the method for increasing the data storage efficiency may further comprises selecting a second data set in dependence of the compression policy, and the step of compressing may further comprise compressing a second data set by making the granularity coarser of the second data set in dependence of the compression policy.

By making the granularity coarser of the second data within the method for increasing the data storage efficiency may further comprise converting the second data set into one or more representations of said data set in one or more data dependent intervals.

Determining the amount of free space in the data repository, and determining to increase the amount of free space in dependence of a free space requirement of the data repository, may further be performed in the method for increasing the data storage efficiency of a data repository

Another object of the present invention is to provide an efficient arrangement for compression of data of a database.

This object is solved by a data processing arrangement for increasing the data storage efficiency of a data repository comprising at least two databases. This data processing

arrangement comprises an information obtaining unit that is arranged to obtain information at least related to the age of the databases, a selection unit that is arranged to select at least a first database according to decreasing age of the databases, and a processing unit, which is arranged to compress at least a first data set of at least the first database by making the granularity coarser of at least the first data set.

The information obtaining unit of the data processing arrangement may further be arranged to obtain information related to a compression policy, the selection unit may further be arranged to select the first database in dependence of the compression policy, and the processing unit may further be arranged to apply the compression policy to compress at least the first data set.

The processing unit of the data processing arrangement may further be arranged to convert at least the first data set into one or more representations of said data set in one or more data dependent intervals.

The selection unit of the data processing arrangement may further be arranged to select a second data set in dependence of the compression policy, the processing unit may further be arranged to compress the second data set by making the granularity coarser of the second data set in dependence of the compression policy, by converting the second data set into one or more representations of the second data set in one or more data dependent intervals.

The processing unit of the data processing arrangement may further be arranged to increase the amount of free space in dependence of a free space requirement of the data repository and the amount of free space in the data repository.

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

BRIEF DESCRIPTION OF THE DRAWINGS

In order to explain the invention and the advantages and features thereof, in more detail, the preferred embodiments will be described below, where references is made to the accompanying drawings, in which

Figure 1 presents a schematic illustration of an arrangement according to some embodiments;
Figures 2-4 present illustrations of method steps according to some embodiments; and
Figures 5a-5d present an illustration of examples of data compression according to some embodiments.

DETAILED DESCRIPTION

A few ways to decrease the size of large data sets were thus described above. Using any one of these ways, the age of the data of the data sets was not taken into account. The age of the data can therefore not affect the way to decrease the size of the data sets.

Herein, within the methods for increasing the efficiency of a data storage related to a data repository, the age of the data of databases is taken into account. Data having different age may thus be treated differently.

In order to describe at least some embodiments, reference is now given to figure 1, presenting a schematic illustration of an arrangement 100 for increasing the data storage efficiency of a data repository.

The arrangement 100 may comprise a database information obtaining unit 102, a selection unit 104, a processing unit 106, a control unit 108 and an input unit 110. As shown in figure 1 the database information obtaining unit 102 may be connected to the selection unit 104. The database information obtaining unit 102 and the selection unit

104 may moreover be connected to the processing unit 106. All three units may in addition be connected to the control unit 108, according to some embodiments. In addition, the selection unit 104 may further be connected to an input unit 110.

The arrangement 100 may also be connected to a temporary storage 112, by way of the processing unit 106 and the control unit 108 of the arrangement, being connected to the temporary storage 112.

The arrangement 100 and the temporary storage 112 may be connected to a data repository 114, where the processing unit 106 and the database information obtaining unit 102 of the arrangement 100 may be connected to the data repository 114.

In order to further describe the arrangement 100, reference is given to figure 2 presenting method steps for increasing the data storage efficiency according to some embodiments upon receiving data be added to a data repository.

Figure 2 shows method steps in a flowchart illustrating a few embodiments.

In step 202, the data to be added are received. These data may an amount of data comprising a plurality of data sets, or may alternatively be a batch of data, as indicated above.

This step may be followed by the step of determining whether to compress the received data or not, step 204. There may be reasons to compress the data, such as to reduce the size of the data. However, there may also be reasons not to compress the data, one of which may be that the data already are compressed. Another reason not to compress the data may be related to the age of the data. One example is that the data may be considered too new to compress. Alternatively, the data may not be well suited for initially being compressed.

If it is determined in step 204 to compress the data, this step is followed by step 206, compressing the received data. This step of initially compressing the received data may mainly serve as an initial step to decrease the size of the data, without significantly losing any data.

As will be explained down below, the step of compressing repository data comprises more features and may be performed in dependence of compression parameters, in a way such that compressing repository data may well be considered more central to the embodiments, than the step of initially compressing received data.

Now, the received data, which may have been compressed in step 206, if answering the interrogation in step 204 in an affirmative way, and which may be uncompressed if answering the interrogation in step 204 in a negative way, may be temporarily stored in the temporary storage 112, in step 208, according to a some embodiments.

In the following step, step 210, the size of the temporarily stored received data is determined. This step may be performed by the temporarily storage 112, or may be performed by the processing unit 106 communicating with the temporarily storage 112.

The subsequent step may be the step of obtaining information related to the data from the data repository 114, step 212. According to some embodiments this step may be performed by the database information obtaining unit 102 of the arrangement 100. In this step the database information obtaining unit 102 may, for instance, obtain information about the available free space in the data repository. The database information obtaining unit 102 may also obtain information about the age of the respective database in the data repository 114.

It can be mentioned that the age of a database may refer to the time that has lapsed since the database was added to the data repository, according to some embodiments. The age of a database may alternatively refer to the age of the data in the database, according to some other embodiments.

In step 214 it may thus be determined whether or not the free repository space is sufficient to store the received data. This determination may be performed by the processing unit 106 using the free space information of the data repository 114 as obtained by the database information obtaining unit 102.

If it is determined that the free repository space is sufficient to store the received data, that is if the interrogation in step 214 is answered in an affirmative way by the processing unit 106, the following step is step 220, retrieving the received data from the temporarily storage 112 such that step 222 can be performed, storing received data in data repository 114.

However, in the case it is determined by the processing unit 106 that the free space of the data repository 114 is not sufficient to store the received data, as stored in the temporary storage 112, in step 214, the step of obtaining selecting of compression policy is performed by the selection unit 104 in step 216. This selection may be performed by using input information, as received from the input unit 110, which may be connected to the selection unit 104, as shown in figure 1

As the compressing policy to apply may now be selected, the step of compressing the repository data by using the selected compression policy, can thus be performed in step 218.

As will be explained in more detail below, in connection with figure 3, the step of compressing the data repository is performed in dependence of the age of the database to compress.

Subsequent to the step of compressing, the following steps 220 and 222, retrieving received data and storing received data in data repository, respectively, are performed in a way at least similar to one that is described above.

Above was thus described a method for increasing the efficiency of a data storage capacity, in connection to receiving new data to be added to the data storage, represented by the data repository.

In addition to the method as described above, there will also be described a method for increasing a data storage efficiency of a data repository, wherein the method may be executed to secure a make free a certain amount of space in the data repository. This method will thus not explicitly comprise increasing the data storage capacity of a data repository upon receiving data to be added to said data repository.

Figure 3 now presents this additional method steps for increasing the efficiency of a data storage capacity, according to at least some embodiments.

This method may start with the step of determining the amount of free space in the data repository, step 302. This stage may be performed due to that data have been added to the data repository at some stage, for instance following the method steps according to figure 2. For this reason the free space available in the data repository may be such that a free space requirement of the database can not be fulfilled. Alternatively, this step may be performed following obtaining a request for more free space available in the data repository. These reasons are thus two examples of reasons to perform the method steps as illustrated in figure 3.

Step 302 may be performed by the selection unit 104, using information about the database as obtained from the database information obtaining unit 102.

Alternatively, the database information obtaining unit 102 may obtain information about the available free space from the data repository, where this step would be performed in the database information obtaining unit 102.

The subsequent step is to determine whether or not the amount of free space fulfils the requirements of free space of the data repository, in step 304. This step may be

performed by the selection unit 104, which may have access to data repository requirements.

If it is determined that the amount of free space does not fulfil the free space requirement, that is, if the interrogation in step 304 is answered negatively, the following step is the step of obtaining selection of compression policy, step 306, which may be performed by the selection unit 104. This step is similar to step 216, as described above.

Having obtained selection of the compression policy in step 306, the next step to be performed is compressing data repository data using the obtained compression policy, in step 308. This step may be performed by the processing unit 106. This step corresponds to step 218, as discussed above.

Having compressed the data repository data the method may end in step 310, according to some embodiments.

However, in the case the amount of free space fulfils the requirement in step 304, the step following step 304 may be ending the method in step 310.

According to an alternative embodiment, this method may be performed at regular intervals by using a trigger, triggering the method steps as illustrated in figure 3 at regular intervals.

Thus the method may for example be performed after a database has been added to the data repository, or be performed at regular intervals, thereby increasing the amount of free space available.

This method may therefore be considered to be a maintenance method that may be performed to secure that, for example, 10% of the total space is free. This free space

corresponding to the 10% can thus be used for adding more data in the form of a database to the data repository.

In the method steps as illustrated in figures 2 and 3, the step of compressing data in the data repository using selected compression policy may well be considered to be the central step of the methods, according to at least some embodiments. For this reason, the attention is now focused upon this step for which further method steps are illustrated in figure 4, presenting method steps for increasing the data storage efficiency of a data repository, as such.

According to some embodiments this method may start by step 402, obtaining indexing of the databases according to age of the data. This step may be performed by the selection unit 104, by using database information as obtained from the database information obtaining unit 102. The selection unit 104 may thus arrange the databases present in the data repository 114 according to the age of the data, in order to enable taking the age of the data of the databases into account when compressing the databases.

Thus, by using the information at least related to data in the data repository, step 402 may be performed. By indexing the databases according to age, the databases may be associated with an order number, which may be used when selecting one database out of a plurality of databases of data repository.

According to some embodiments compression of data may be performed following a compression policy. Compression may however be performed without applying a specific compression policy according to alternative embodiments. Nevertheless, the age of the databases still has to taken into account when selecting which database to compress.

In the case a compression policy is applied, information about which policy to apply may be obtained from earlier received information, such as the information as received

in step 216 or 306, obtaining selection of compression policy, or may be obtained at other instances, possibly via the input unit 110 connected to the selection unit 104.

Now, having indexed the databases according to age, the step of selecting a database dependent on the obtained selection of compression policy and dependent on the age of the data of the data sets, is performed in step 404.

According to at least some embodiments the step of selecting a database is performed according to decreasing age of the databases. In this respect the database that is oldest may be selected at first. As will be described below, the selection may alternatively be performed according to another age related parameter, such as selecting the least recently compressed database.

One example of a compression policy is the so called, first in first out (FIFO) compression policy. In short, this policy states that the database that was first added to the data repository, of all added databases in the data repository, is the one to be compressed first. This implies that this database will be fully compressed until the maximum compression level has been reached, before a second database, which then is the second oldest of the databases in the data repository, will be compressed at all in the data repository.

A second example of a compression policy is the so called, "Round Robin" compression policy.

Applying this policy, the age of the individual databases is also taken into account. However, in this case the databases are compressed one after the other in a ring, starting from the one that was the first to be compressed of all participating databases in the data repository. Having compressed this first database once, for instance such that the degree of compression, which may be measured by the compression level, is incremented one step, the second database to be compressed applying the Round Robin compression policy would be the database that initially was the second to be compressed. The third

database to be compressed would therefore be the database having the third oldest compression.

Using Round Robin, each database may be indexed or marked according to the time that has elapsed since it was last compressed. Upon compressing the databases, this index or marker may thus be used to select the database to compress. Using the Round Robin compression scheme each database is typically not fully compressed, before the next database to be selected to be compressed, is actually compressed. This is in contrast to the FIFO compression policy, in which the first database typically is compressed fully, reaching a maximum compression level, before a second database is selected and can be started to be compressed.

In addition, according to some embodiments other compression policies may also be applied, considered they do take the age of the data into account when determining the order to compress the databases.

Having selected a database to compress by the selection unit 104, in step 404, the following step is the step of determining the compression level needed for the selected database, step 406. As mentioned above, the compression level is one measure of the degree of compression for a database.

This step may be performed in the processing unit 106, by using information from the data repository, as obtained from the database information obtaining unit 102, and possibly information regarding any requirements for free space in the data repository, possibly received from the input unit 110 via the selection unit 104.

As was described above, compression of databases in the data repository may be performed to secure that a certain amount of free space is available, such that data that may have been received can be added to the data repository. Another reason to compress data in the data repository can be to secure that a degree of free space is available, for example, after having added received data, after having modified the data

in the repository or possibly after having altered the data storage capacity of the data repository itself.

As will be explained below various compression levels can be defined for each database, corresponding to the degree of compression of the database and indicating the amount of free space that has been achieved.

In the step of 406, the processing unit 106 may calculate the compression level needed. Alternatively, the processing unit 106 may estimate the compression level needed in order to achieve a certain amount of free space. Such an estimation may be based on data obtained from compression of other databases that were earlier compressed.

In step 408, it is determined whether or not the needed compression level of the selected database is higher than or equal to the maximum compression level of said database. This step may be performed by the processing unit 106. In the case the needed compression level is higher than or equal to the maximum compression level of the database, a second database has to be selected and compressed in addition to having compressed the first database in order to achieve the free space requirement. It is thus determined whether or not there is a need to compress a further database in addition to compressing the selected first database.

The processing unit 106 may have access to compression history data from other databases, such that it can perform an estimation of how much the data in the selected database can be compressed.

As each database may contain various kinds of data comprising data fields having numbers, strings, signs, etcetera, the maximum compression level of the database may be estimated by using compression history data, and used in the determination in step 408.

In the case it is determined that the needed compression level is not larger than the maximum compression level of the selected database, that is the case in which the interrogation in step 308 is answered negatively, the next task may be to compress the data sets of the selected database accordingly in step 414, compressing the data sets of that database by making the granularity coarser of said data sets until the needed compression level is achieved.

Making the granularity coarser of the data of the database, will be described and exemplified down below. However, an introduction of the concept of granularity is here included in order to avoid clarify the concept of granularity.

Increasing an efficiency of a data storage such as a data repository, as discussed herein, relates to transforming data, such as for instance, data values from numbers to representations in intervals, ranges or groups. Instead of comprising detailed numbers, transformed or converted data comprise information in the form of representations of that the detailed numbers are positioned in certain ranges, which have to be individually defined dependent on the data to be compressed and on the level of compression.

Detailed uncompressed data can be viewed upon as data divided in a large number of groups, one for each data value. The information content of the data is thus not affected by the grouping of the data since all numbers still are accessible. This division of data is defined to correspond to a considerably fine granularity of the data.

By decreasing the number of groups of intervals, the data are divided in wider and wider groups or intervals, for which reason the granularity is made coarser. A coarse granularity of the data corresponds to a division of the data in significantly wide groups or intervals.

Since the granularity is made coarser in step 414, the compression of the data sets of the databases thus results in a coarser definition of the data.

In the case that the needed compression level is larger than or equal to the maximum compression level of the selected database in step 408, the current method steps continue with step 410, determining whether or not the current compression level is smaller than the maximum compression level of the selected database. This step may be performed by the processing unit 106 and is performed in order to determine whether the current database can be further compressed or not.

In the case the current compression level is smaller than the maximum compression level of the selected database, in the case of an affirmative answer to the interrogation in step 410, the selected database can be further compressed until the maximum compression level is achieved in step 412, compressing data sets of database by making the granularity coarser of said data sets until maximum compression of database is achieved. This step may also be executed in the processing unit 106.

After step 412, and after step 410 in the case the current compression level is determined to be larger than the maximum compressing level of the selected database, the subsequent step is the step of obtaining selection of database dependent on the selected compression policy and the age of the data in the databases, step 404. This step and the following steps are thus performed until the free space requirement is fulfilled.

In order to further clarify the step of making the granularity coarser of data sets in the databases and in order to give examples how such a step may be executed, reference is now made to figures 5a-5d, illustrating examples of how log data may look like at various compression levels.

Starting with figure 5a, illustrating log data that are uncompressed, four columns are illustrated showing data in the form of time of start, duration, type of call and direction of the call. The duration column comprises data in the form of values in minutes.

In figure 5b the log data of figure 5a are now illustrated with the only exception that the data have been compressed one step to reach higher compression level. This

compression level is here called compression level 1. It is shown that the time column is unchanged.

It can be noted that the values of the duration parameter have been given a 16 bit space of the data repository, which is designed to cover the 384 value of the parameter.

The type parameter is however compressed by representing any voice call by an "0", any SMS by a "1", and any other calls by "2". It is thus required a 2 bit space, to encompass these three alternatives.

Similarly, the direction parameter is compressed by using 2 bit, letting incoming calls be represented by a"0", outgoing calls by "1" and other calls by a "2".

Further compression of the log data according to figure 5a, may result in the compression level 2. In addition to the compression as performed for figure 5b, the duration parameter is compressed using 2 bit space, with the following representations, letting "0-99" =0, "100-199" = 1, "200-299" =2, and "300-399" =3. Also the direction parameter has been compressed further, now comprising 1 bit with the representations "in"=0, and "other"=1.

Referring to figure 5d, it is shown that the time is still uncompressed, whereas the duration parameter now has been given 1 bit with the representations "0-199"=0, "200-399"=1. The type and direction parameters are compressed as in figure 5c.

The concept of compressing the data according to the embodiments as presented herein, is to make the granularity coarser of the data upon compression. Uncompressed log data may be strings or numbers having practically any combination of letters and signs and/or symbols.

By introducing a granularity of data, that is a division of data into groups, ranges or intervals, data can be converted into a representation of an occurrence in said groups, ranges or intervals.

A singular data value is herewith converted into an occurrence within a range of values, for which reason the space required for storing the data may be decreased.

Data comprising strings of data may be compressed by converting each string to a representation of the strings, as indicated in figure 5b in the direction field wherein a "1" represents "outgoing", and "0" represents "incoming".

According to some embodiments the relative frequency of specific strings in data could serve as basis to representations of said strings in compressed data.

By gradually making the granularity of data coarser by converting at least one of ordinal, interval and ratio values into values that represent successively larger ranges, a higher degree of compression level may thus be achieved.

Also, this may be achieved by reducing the number of possible distinct interval classes for nominal values.

By treating stored data as separate databases dependent of the age of the data, the databases may be compressed as a response to received new data or as a measure to secure a certain amount of free space in the data repository storing the databases. The data will thus be processed differently depending on how old the data is, while still enabling inspection of the data without having to uncompress it.

The process of compressing data may be repeated until each original data value is represented by 1 bit that relates to a corresponding value range. This was exemplified in figures 5a-5d, which was described above.

As was earlier described above, the method for increasing the efficiency of data storage, stores the transformed data in intervals, ranges or groups, in a way such that the data is ready to be retrieved without the need to unpack the data. The idea of unpacking data is therefore superfluous to retrieve the compressed data.

A lookup table is however required to consult in order to obtain information about what each interval, range or group represent. There is thus only a minor extra expense reading the representations in the intervals, ranges or groups, which corresponds to looking up the table. However, since looking up in the table only has to be done once when retrieving data, it is believed that the latency of having to look up what each division represents, is negligible.

It should also be mentioned that the concept of compressing log data is a lossy compression, in respect of that details of data may be lost when the data are converted to representations in intervals, groups or ranges.

The compressing of data may be configurable in a way such that when the age of the increases, the data may be compressed further until a maximum compression level is reached and no more compression can be carried out. One reason for not compressing the data further is that all data values are represented by 1 bit. Another reason may the introduction of a loss threshold, which sets a maximum acceptable loss of data for each database. This threshold may thus define an upper limit of the compression level of the data.

One example of using compression of data in dependence of the age of the data, may be an automated learning functionality for which batches of data can be processed. A relevance parameter may be defined, which reflects the weight of the data. New data have a higher weight than old data, and uncompressed data have a higher weight than compressed data. As the age of the data is increased the relevance parameter may thus be decreased. Similarly when data are compressed the relevance parameter may also be decreased. Data having the highest relevance are thus new uncompressed data.

The learning functionality may thus be defined to pay more attention to data having a high relevance, and to pay less attention to data having a low relevance.

It can be pointed out the embodiments as mentioned above represent examples of embodiments and that these can be varied in many ways.

According to embodiments a distribution analysis can be performed for the data to be compressed, such that the definition of interval ranges can be defined such that intervals that have no representation can be avoided. The intervals may preferably be defined such as that the converted data are spread among the intervals, increasing the usage of the intervals.

According to further embodiments distribution functions could be used to better and more space efficient group data in compressed intervals.

The different embodiments are hence non-limiting examples. The scope of the present invention is only limited by the subsequently following claims.

It can be easily understood that the at least some of the embodiments come with advantages such as:

Compressed data may be quickly accessible without the need to perform a time consuming de-compression. An indexed retrieval of stored data information is thus provided.

The time dependence of the compression enables weighting of data according to age, such that new data, which often are regarded as more important, are given a higher relevance, since older data may be compressed harder.

CLAIMS

1. A method for increasing the data storage efficiency of at least a first database of a data repository (114) comprising at least two databases, the method comprising the steps of:
   - obtaining information at least related to the age of the databases (steps 212, 402)
   - selecting at least a first database in dependence of the age of the databases (step 404), and
   - compressing at least a first data set of at least the first database by making the granularity coarser of at least the first data set (steps 412, 414).

2. The method for increasing the data storage efficiency of a data repository (114), according to claim 1, wherein the step of selecting comprises selecting at least the first database according to decreasing age of the databases (step 404).

3. The method for increasing the data storage efficiency of a data repository (114), according to claim 1 or 2, wherein the step of obtaining information comprises obtaining information related to a compression policy (steps 212, 402), wherein the step of selecting comprises selecting the first database in dependence of the compressing policy (step 404), and wherein the step of compressing comprises applying the compression policy for compressing at least the first data set (steps 412, 414).

4. The method for increasing the data storage efficiency of a data repository (114), according to any one of claims 1-3, wherein making the granularity coarser of at least the first data set, comprises converting at least the first data set into one or more representations of said data set in one or more data dependent intervals.

5. The method for increasing the data storage efficiency of a data repository, according to claim 3 or 4, wherein the step of selecting further comprises

selecting a second data set in dependence of the compression policy (404), and wherein the step of compressing comprises compressing a second data set by making the granularity coarser of the second data set in dependence of the compression policy (steps 412, 414).

6. The method for increasing the data storage efficiency of a data repository, according to claim 5, wherein making the granularity coarser of the second data further comprises converting the second data set into one or more representations of said data set in one or more data dependent intervals.

7. The method for increasing the data storage efficiency of a data repository, according to any one of claims 1-6, further comprising determining the amount of free space (step 212, 302) in the data repository, and determining to increase the amount of free space in dependence of a free space requirement of the data repository (step 214, 304).

8. A data processing arrangement (100) for increasing the data storage efficiency of a data repository (114) comprising at least two databases, comprising:
   - an information obtaining unit (102), arranged to obtain information at least related to the age of the databases,
   - a selection unit (104), arranged to select at least a first database according to decreasing age of the databases, and
   - a processing unit (106), arranged to compress at least a first data set of at least the first database by making the granularity coarser of at least the first data set.

9. The data processing arrangement (100) for increasing the data storage efficiency according to claim 8, wherein the information obtaining unit (102) further is arranged to obtain information related to a compression policy, wherein the selection unit (104) further is arranged to select the first database in dependence of the compression policy, and wherein the processing unit (106) further is arranged to apply the compression policy to compress at least the first data set.

10. The data processing arrangement (100) for increasing the data storage efficiency according to claim 8 or 9, wherein the processing unit (106) further is arranged to convert at least the first data set into one or more representations of said data set in one or more data dependent intervals.

11. The data processing arrangement (100) for increasing the data storage efficiency according to claim 9 or 10, wherein the selection unit (104) further is arranged to select a second data set in dependence of the compression policy, and wherein the processing unit (106) further is arranged to compress the second data set by making the granularity coarser of the second data set in dependence of the compression policy, by converting the second data set into one or more representations of the second data set in one or more data dependent intervals.

12. The data processing arrangement (100) for increasing the data storage efficiency according to any one of claims 8-11, wherein the processing unit (106) is arranged to increase the amount of free space in dependence of a free space requirement of the data repository (114) and the amount of free space in the data repository (114).

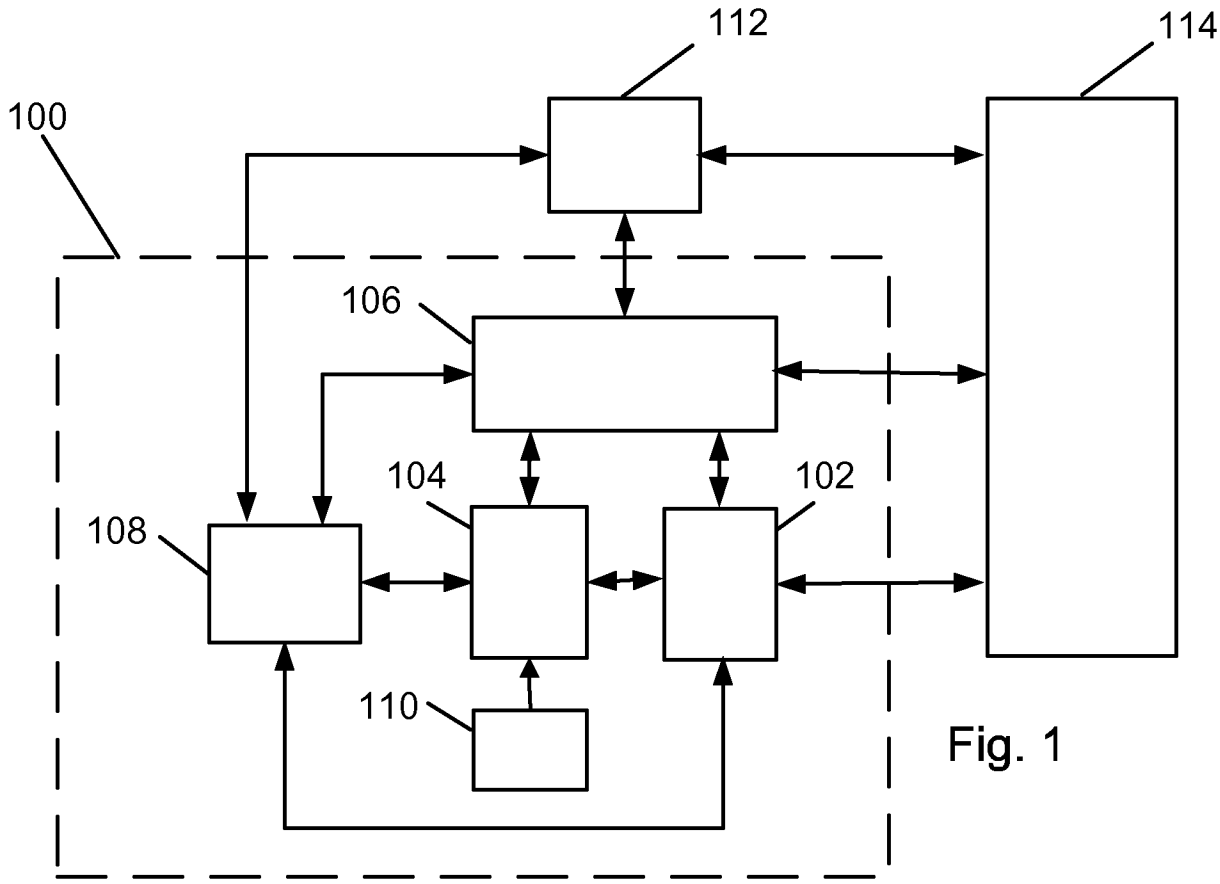
112
114
100
106
104
102
108
110

Fig. 1

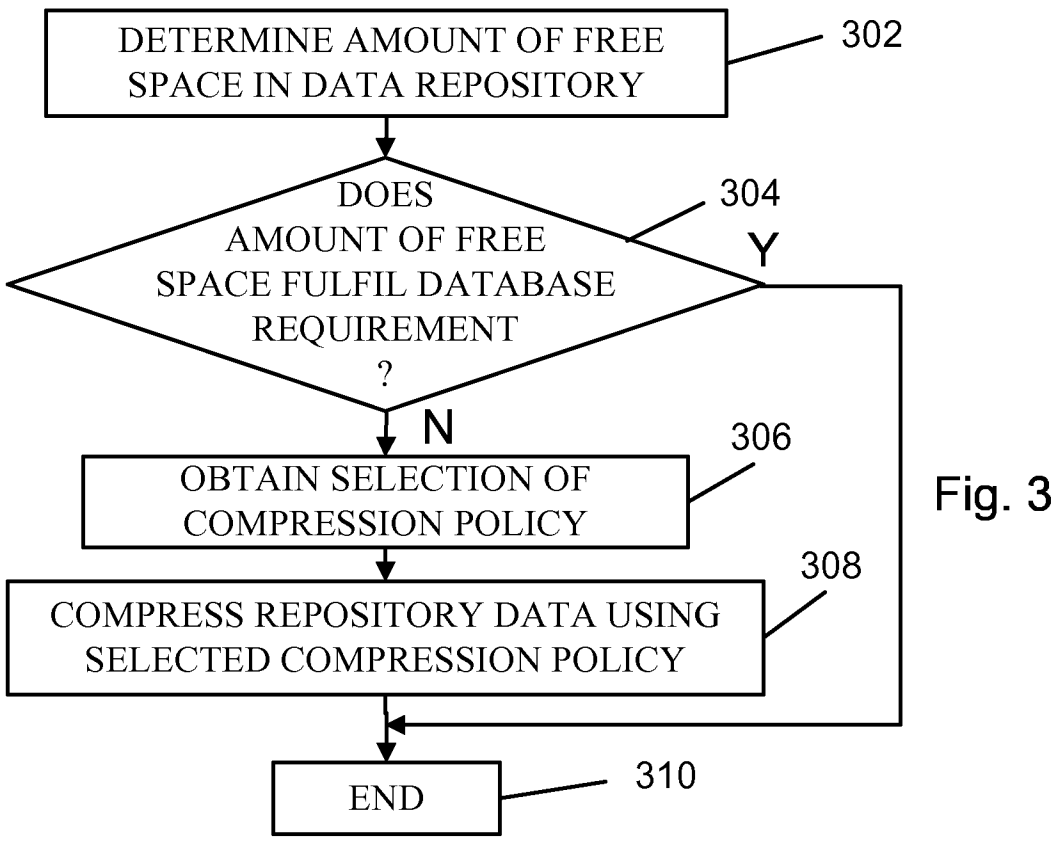
DETERMINE AMOUNT OF FREE SPACE IN DATA REPOSITORY
302
DOES AMOUNT OF FREE SPACE FULFIL DATABASE REQUIREMENT ?
304
Y
N
OBTAIN SELECTION OF COMPRESSION POLICY
306
COMPRESS REPOSITORY DATA USING SELECTED COMPRESSION POLICY
308
END
310

Fig. 3

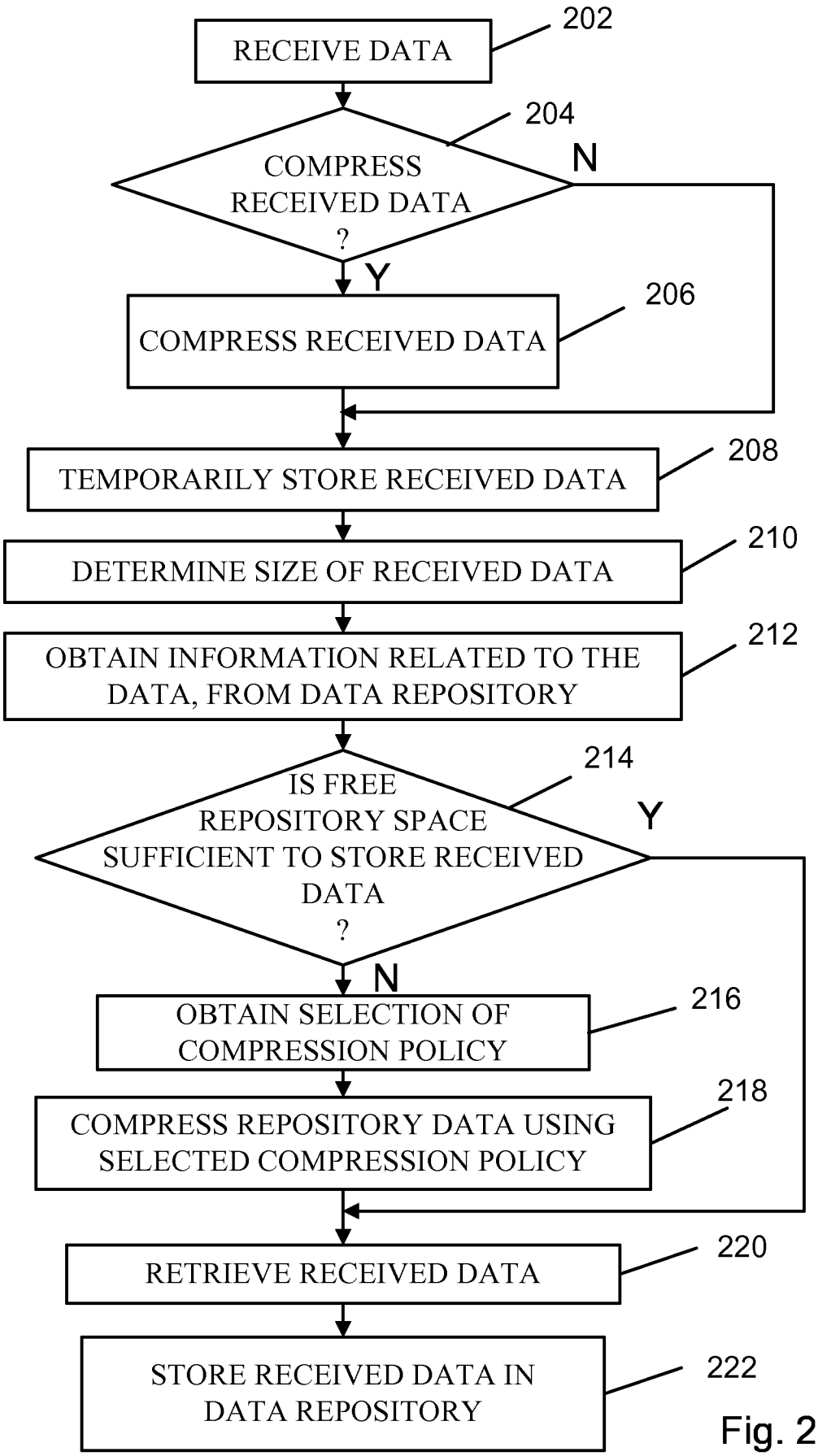
RECEIVE DATA
202
COMPRESS RECEIVED DATA ?
204
N
Y
COMPRESS RECEIVED DATA
206
TEMPORARILY STORE RECEIVED DATA
208
DETERMINE SIZE OF RECEIVED DATA
210
OBTAIN INFORMATION RELATED TO THE DATA, FROM DATA REPOSITORY
212
IS FREE REPOSITORY SPACE SUFFICIENT TO STORE RECEIVED DATA ?
214
Y
N
OBTAIN SELECTION OF COMPRESSION POLICY
216
COMPRESS REPOSITORY DATA USING SELECTED COMPRESSION POLICY
218
RETRIEVE RECEIVED DATA
220
STORE RECEIVED DATA IN DATA REPOSITORY
222

Fig. 2

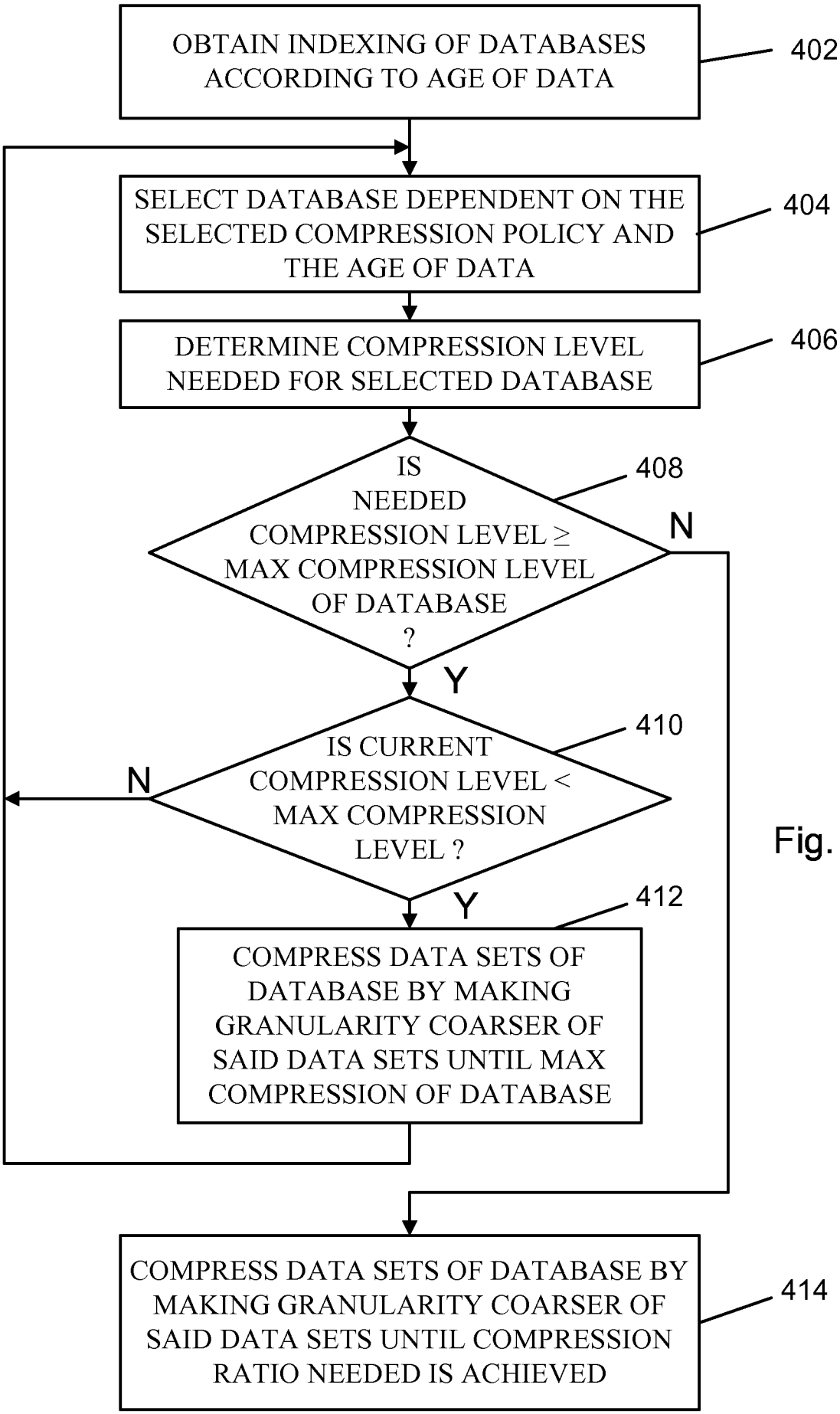
OBTAIN INDEXING OF DATABASES ACCORDING TO AGE OF DATA
402
SELECT DATABASE DEPENDENT ON THE SELECTED COMPRESSION POLICY AND THE AGE OF DATA
404
DETERMINE COMPRESSION LEVEL NEEDED FOR SELECTED DATABASE
406
IS NEEDED COMPRESSION LEVEL ≥ MAX COMPRESSION LEVEL OF DATABASE ?
408
N
Y
IS CURRENT COMPRESSION LEVEL < MAX COMPRESSION LEVEL ?
410
N
Y
COMPRESS DATA SETS OF DATABASE BY MAKING GRANULARITY COARSER OF SAID DATA SETS UNTIL MAX COMPRESSION OF DATABASE
412
COMPRESS DATA SETS OF DATABASE BY MAKING GRANULARITY COARSER OF SAID DATA SETS UNTIL COMPRESSION RATIO NEEDED IS ACHIEVED
414

Fig. 4

Call log (No compression)			
Time	Dur.	Type	Direction
040910:2101	384	Voice all	Outgoing
040911:1010	23	Voice all	Incoming
040912:0004	0	SMS	Incoming
040913:1809	120	Voice call	Outgoing

Fig. 5a

Call log (Compression level 1)			
Time	Dur.	Type	Direction
040910:2101	384	0	1
040911:1010	23	0	0
040912:0004	0	1	0
040913:1809	120	0	1

Fig. 5b

Call log (Compression level 2)			
Time	Dur.	Type	Direction
040910:2101	3	0	1
040911:1010	0	0	0
040912:0004	0	1	0
040913:1809	1	0	1

Fig. 5c

Call log (Compression level 3)			
Time	Dur.	Type	Direction
040910:2101	1	0	1
040911:1010	0	0	0
040912:0004	0	1	0
040913:1809	0	0	1

Fig. 5d

INTERNATIONAL SEARCH REPORT

International application No
PCT/EP2008/051202

A. CLASSIFICATION OF SUBJECT MATTER
INV. G06F17/30

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 practical, 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 6 513 065 B1 (HAFEZ AMR [US] ET AL) 28 January 2003 (2003-01-28) abstract column 6, line 11 - column 9, line 55 column 11, line 56 - column 17, line 59 ----- -/--	1-12

[X] Further documents are listed in the continuation of Box C.

[X] 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 document but published on or after the international filing date

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

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

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

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

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

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

"&" document member of the same patent family

Date of the actual completion of the international search	Date of mailing of the international search report
25 June 2008	09/07/2008

Name and mailing address of the ISA/	Authorized officer
European Patent Office, P.B. 5818 Patentlaan 2 NL - 2280 HV Rijswijk Tel. (+31-70) 340-2040, Tx. 31 651 epo nl, Fax: (+31-70) 340-3016	Dumitrescu, Cristina

Form PCT/ISA/210 (second sheet) (April 2005)

INTERNATIONAL SEARCH REPORT

International application No
PCT/EP2008/051202

C(Continuation). DOCUMENTS CONSIDERED TO BE RELEVANT

Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
X	ALTIPARMAK F; CHIU D; FERHATOSMANOGLU H: "Incremental quantization for aging data streams" 2007 7TH IEEE INTERNATIONAL CONFERENCE ON DATA MINING WORKSHOPS, [Online] 28 January 2007 (2007-01-28), - 31 October 2007 (2007-10-31) pages 527-532, XP002485586 Omaha, NE, USA Retrieved from the Internet: URL:http://www.ieeexplore.ieee.org/iel5/4476629/4476630/04476718.pdf?tp=&isnumber=4476630&arnumber=4476718> [retrieved on 2008-06-25] abstract page 527 page 530 - page 532	1-12
X	US 2006/059172 A1 (DEVARAKONDA MURTHY V [US]) 16 March 2006 (2006-03-16) abstract paragraph [0004] paragraph [0009] - paragraph [0012] paragraph [0024] - paragraph [0030]	1-12
A	US 2002/069324 A1 (GERASIMOV DENNIS V [US] ET AL) 6 June 2002 (2002-06-06) abstract paragraph [0063] - paragraph [0067] paragraph [0109] - paragraph [0116]	1-12

INTERNATIONAL SEARCH REPORT

Information on patent family members

International application No
PCT/EP2008/051202

Patent document cited in search report	Publication date	Patent family member(s)	Publication date
US 6513065 B1	28-01-2003	NONE	
US 2006059172 A1	16-03-2006	NONE	
US 2002069324 A1	06-06-2002	NONE	