



(51) International Patent Classification:

G06F 17/30 (2006.01) *G06F 11/14* (2006.01)
G06F 3/06 (2006.01) *G06F 9/445* (2006.01)

(21) International Application Number:

PCT/EP2015/059394

(22) International Filing Date:

29 April 2015 (29.04.2015)

(25) Filing Language:

English

(26) Publication Language:

English

(71) **Applicant** (for all designated States except US): **HEW-LETT-PACKARD DEVELOPMENT COMPANY, L.P.** [US/US]; 11445 Compaq Center Drive West, Houston, 77070 (US).

(72) Inventors; and

(71) **Applicants** (for US only): **DONAGHY, Dave** [GB/GB]; Longdown Avenue, Stoke Gifford, Bristol BS34 8QZ (GB). **BUTT, John** [GB/GB]; Longdown Avenue, Stoke Gifford, Bristol BS34 8QZ (GB).

(74) **Agents**: **STÖCKELER, Ferdinand** et al.; Radlkoferstr.2, 81373 München (DE).

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

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

Published:

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

(54) Title: DEDUPLICATION PROCESS

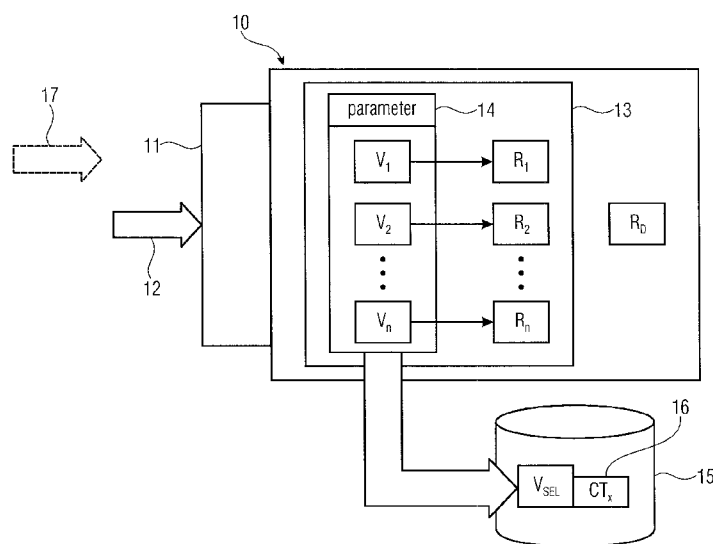


FIG 1

(57) **Abstract:** A computing device comprises an input to receive input data and a processor to perform a simulation of a deduplication process to process the input data, wherein the deduplication process comprises a modifiable parameter and wherein a processing result depends on the value of the modifiable parameter. The processor is to perform the deduplication process a plurality of times each time with a different value of the modifiable parameter and to obtain a result for each value. The processor is to select the value associated with a result closest to a desired result as the value to be used in processing the input data. The processor is to store the selected value of the modifiable parameter associated with a content type identifier in memory, compare the content type of other input data to the stored content type identifier, and to select the stored parameter value for processing the other input data if the content type of the other input data corresponds to the stored content type identifier.

DEDUPLICATION PROCESS

BACKGROUND

[0001] Computer systems may include computing devices configured to perform processes such as data processing. The data processing may include data and parameters that may be used to track different processes of the system.

BRIEF DESCRIPTION OF THE DRAWINGS

[0002] Examples will now be described by way of non-limiting example only with reference to the accompanying drawings, in which:

[0003] Figure 1 is a simplified illustration of a computing device according to an example,

[0004] Figure 2 is a simplified illustration of a computing device according to a further example,

[0005] Figure 3 is a simplified illustration of a part of a computing device according to a further example,

[0006] Figure 4 is a simplified illustration of another part of a computing device according to a further example,

[0007] Figure 5 is a simplified illustration of a computing device according to a further example,

[0008] Figure 6 is a block diagram outlining a method for processing input data according to an example,

[0009] Figure 7 is a flow diagram outlining a method for processing input data according to an example, and

[0010] Figure 8 is a simplified illustration of a non-transitory machine-readable storage medium encoded with instructions executable by a processing resource of a computing device to perform a method according to an example.

DETAILED DESCRIPTION

[0011] The examples that will be described in the following may be used to perform a deduplication process. A deduplication process may be used, for instance, to identify redundant data, i.e. duplicates, and to eliminate the identified redundant data before this redundant data is written on a data storage medium.

[0012] For example, a deduplication process may comprise a chunker process that is to split input data into chunks before storing this data on a data storage medium. The chunks may comprise different sizes. With the assumption that input data may contain inherent duplication, the chunker process is intended to produce many similar chunks that may comprise the same parts or segments of the input data. Accordingly, these chunks contain redundant data.

[0013] For example, it could be analyzed on byte-level, which chunks may provide for a large repetition rate. A high amount of data reduction can be provided in that similar chunks containing redundant data may be eliminated. The content of the eliminated chunks may be referenced with pointers which use less space on the data storage medium than the referenced and eliminated bit sequence would do.

[0014] When the input data is recalled, it may be reconstructed from its individual parts and from the referenced parts that it might have in common with other input data.

[0015] A chunker process may have two modifiable parameters, namely a maximum chunk size and a minimum chunk size. In between these two sizes, a chunk boundary may be selected in order to obtain a desired result.

[0016] Regarding a deduplication process, a desired result may, for instance, be a desired chunk size. A desired chunk size may be a size of a chunk that provides for a high deduplication ratio. A deduplication ratio is the ratio of the size of the input data to the size actually used at the storage medium to store the input data.

[0017] Examples described herein provide a computing device comprising an input to receive input data and a processor to perform a deduplication process to process the input data. The deduplication process may comprise a modifiable parameter and a processing result may depend on the value of the modifiable parameter. The processor is to perform a simulation of the deduplication process, wherein the deduplication process is

performed a plurality of times, each time with a different value of the modifiable parameter, and to obtain a result for each value. The processor is to select the value associated with a result closest to a desired result as the value to be used in processing the input data. The processor is further to store the selected value of the modifiable parameter associated with a content type identifier in memory, compare the content type of other input data to the stored content type identifier, and to select the stored parameter value for processing the other input data if the content type of the other input data corresponds to the stored content type identifier. Thus, it is possible to simulate the effects that a choice of a respective one of the different values may have on the deduplication process. It may be possible to simulate these effects before the input data is actually processed in the deduplication process. Accordingly, the result of the deduplication process, depending on the selected value, can be predicted before the input data is actually processed in the deduplication process.

[0018] Furthermore, content-type specific processing of the input data in the deduplication process may be provided and performed. That is, the deduplication process may be performed in dependence of the content type of the respective input data. For example, if it is determined that the input data comprises a certain content, such as music, specific parameter values that may be well suited to process music data may be used in the deduplication process. These specific values may be stored, and thus be available, in the memory. They may be available in the memory, for instance, if a content-specific data type, such as e.g. music data, has been previously processed and the previously selected value for that data content type has been stored, or if a preset of values that may be well suited for processing that data content type, e.g. music data, may have been stored previously.

[0019] In an example, the deduplication process may comprise more than one modifiable parameter, which may in the following also be referred to as a plurality of modifiable parameters or as a set of modifiable parameters.

[0020] According to an example, the modifiable parameter may have a number of possible values and the processor may be to perform the deduplication process with each one of the possible values. Thus, the computing device according to this example may perform the deduplication process with a large number of different values, directly at the installation site or the implementation site of the deduplication process, however, in a short time. In other words, a large volumetric quantity of differing values and value

combinations may be attempted before actually processing the input data in the deduplication process.

[0021] According to a further example, the deduplication process may comprise a plurality of modifiable parameters, wherein the processing result depends on the value of each of the modifiable parameters. The processor may be to perform the process a plurality of times each time with a different value of each of the modifiable parameters and to obtain a result for each value of each modifiable parameter. The processor may be to select a value of each of the modifiable parameters associated with a result closest to a desired result as the value of each of the modifiable parameters to be used in processing the input data. Accordingly, more than one modifiable parameter may be available, wherein each of the modifiable parameter may comprise one or more values. A desired result may be achieved by using different combinations of values for each of the modifiable parameters. Even though the effort for performing the deduplication process may increase with the number of possible combinations, the computing device may provide for an accurate processing in a short time. The computing device may also perform the deduplication process directly at the installation site or the implementation site. A plurality of modifiable parameters may also be referred to as a set of modifiable parameters in the following.

[0022] Other examples described herein may provide a method, implemented by a computing device, the method comprising receiving input data by the computing device and performing, by the computing device, a deduplication process to process the input data, wherein the deduplication process may comprise a modifiable parameter and wherein a processing result may depend on the value of the modifiable parameter. The method comprises performing, by the computing device, a simulation of the deduplication process by performing the deduplication process a plurality of times, each time with a different value of the modifiable parameter, and obtaining a result for each value. The method comprises selecting, by the computing device, the value associated with a result closest to a desired result as the value to be used in processing the input data in a deduplication process to be performed subsequent to the simulation of the deduplication process. The method further comprises storing the selected value of the modifiable parameter associated with a content type identifier in memory, comparing the content type of other input data to the stored content type identifier, and selecting the stored parameter value for processing the other input data if the content type of the other input data corresponds to the stored content type identifier. Thus, it is possible to simulate the effects that a choice of a respective one of the different values may have on the deduplication

process. It may be possible to simulate these effects before the input data is actually processed in the deduplication process. Accordingly, the result of the deduplication process, depending on the selected value, can be predicted before the input data is actually processed in the deduplication process.

[0023] Figure 1 shows an example of a computing device 10. The computing device 10 comprises an input 11 to receive input data 12. The input data 12 may be provided, for instance, as a contiguous data stream.

[0024] The computing device 10 comprises a processor 13 to perform a deduplication process to process the input data 12. The deduplication process comprises a modifiable parameter 14. The processing result $R_1, R_2, \dots, R_n$ of the deduplication process depends on the value $V_1, V_2, \dots, V_n$ of the modifiable parameter 14.

[0025] Prior to performing the actual deduplication process, the processor 13 is to perform a simulation of the deduplication process, wherein the deduplication process is performed a plurality of times, each time with a different value $V_1, V_2, \dots, V_n$ of the modifiable parameter 14, and to obtain a result $R_1, R_2, \dots, R_n$ for each value $V_1, V_2, \dots, V_n$.

[0026] The processor 13 is to select the value $R_1, R_2, \dots, R_n$ associated with a result $R_1, R_2, \dots, R_n$ closest to a desired result R_D as the value to be used in processing the input data 12.

[0027] In other words, the computing device 10 receives input data 12 via the input 11. The input data 12 is processed by the processor 13, wherein the processor 13 performs a deduplication process.

[0028] The deduplication process comprises at least one modifiable parameter 14. The modifiable parameter 14 may comprise several values $V_1, V_2, \dots, V_n$. The processor 13 may repeatedly perform the deduplication process, each time with a different one of the available parameter values $V_1, V_2, \dots, V_n$. Depending on the value $V_1, V_2, \dots, V_n$ that has been selected for performing the deduplication process, a respective result $R_1, R_2, \dots, R_n$ may be obtained. By processing the input data 12 several times, each time with a different value $V_1, V_2, \dots, V_n$, a simulation of the deduplication process may be performed.

[0029] For example, as shown in Figure 1, if the value V_1 is selected for performing the deduplication process, the respective result that may be obtained is result

R_1 . If the value V_2 is selected for performing the deduplication process, the respective result that may be obtained is result R_2 . Stated in more general terms, if the value V_n is selected for performing the deduplication process, the respective result that may be obtained is result R_n , wherein the letter 'n' is a natural number and represents the index of a plurality of values and results, respectively.

[0030] As mentioned before, the processor 13 may perform the deduplication process several times with the input data 12. Each time, the processor 13 may use a different value $V_1, V_2, \dots, V_n$ when performing the deduplication process. For each value $V_1, V_2, \dots, V_n$, a respective result $R_1, R_2, \dots, R_n$ is obtained. Thus, the processor 13 is to perform a simulation of the deduplication process when processing a current input data 12.

[0031] The processor 13 may compare the obtained results $R_1, R_2, \dots, R_n$ with each other. The processor 13 may also compare the obtained results $R_1, R_2, \dots, R_n$ with a desired result R_D .

[0032] A desired result R_D is the result that may be desired by the person or machine that initiates or performs the deduplication process. In other words, the desired result R_D may be the desired output of the deduplication process. In a deduplication process it may, for instance, be desired to achieve a high deduplication ratio as the desired result.

[0033] The processor 13 may determine that result $R_1, R_2, \dots, R_n$ that is closest to the desired result R_D . The processor 13 may select the corresponding value $V_1, V_2, \dots, V_n$ that yields this closest result. For example, if the second result R_2 may be the result that is closest to the desired result R_D , the corresponding second value V_2 is selected by the processor 13.

[0034] The processor 13 may then use the selected value $V_1, V_2, \dots, V_n$, for example V_2 , to process the current input data 12 subsequent to the simulation and to generate a respective output. In other words, the processor 13 may compare one or more of the simulation results $R_1, R_2, \dots, R_n$ with the respective desired result R_D . For the actual deduplication process, i.e. for the deduplication process that is performed subsequent to the previously performed simulation, the processor 13 may select the respective value that gained the closest result during the simulation.

[0035] Stated yet differently, the processor 13 may perform a simulation of the deduplication process by performing the deduplication process several times, each time with a different value $V_1, V_2, \dots, V_n$. If the processor 13 determines, during or after the simulation, the result $R_1, R_2, \dots, R_n$ that is closest to the desired result R_D , the processor 13 may select the respective value $V_1, V_2, \dots, V_n$ with which the closest result can be achieved. The input data 12 is then processed in the deduplication process with that value that has been selected from the simulation.

[0036] The computing device 10 may further comprise a memory or storage 15. If the processor 13 may select a certain parameter value V_{SEL} , the processor 13 is to store this selected value V_{SEL} together with a content type identifier 16 in the memory 15. The content type identifier 16 may be used to identify the content of the input data 12. For example, if the input data 12 may be a music data stream, then the content type identifier 16 may be used to identify the input data 12 as a music data stream. The content type identifier 16 may be carried as payload within the input data 12, or could be otherwise derived by the computing device 10.

[0037] If input data other than the previously simulated and processed input data 12, for example, input data 17 that may arrive later in time or at a different site, may arrive at the computing device 10, then the processor 13 may use the previously stored and thus already available selected value V_{SEL} for processing the other input data 17, if the content type of the other input data 17 corresponds to the content type 16 associated with the selected value V_{SEL} . Accordingly, a simulation of the deduplication process may be omitted when other input data 17 is processed, which other input data 17 may comprise a similar or the same content as the previously processed input data 12 for which a simulation has been previously performed. In this case, the simulation may be omitted due to the fact that the previously selected value V_{SEL} is already available in the memory 15 and may therefore be used for processing the other input data 17. An example will be described in more detail below, with respect to Figure 2.

[0038] The modifiable parameter 14 has a number of possible values $V_1, V_2, \dots, V_n$ with which it may be possible to perform the deduplication process and the simulation of the deduplication process, respectively. The processor 13 may perform the deduplication process with a selection of the possible values, for example, only for values V_2, V_5 and V_7 . In an example, the processor 13 may perform the deduplication process with each one of the possible values $V_1, V_2, \dots, V_n$. The processor 13 may obtain a respective result $R_1, R_2, \dots, R_n$ for each one of the values $V_1, V_2, \dots, V_n$.

[0039] Accordingly, the processor 13 may perform a full simulation of the deduplication process by using the full set of available values $V_1, V_2, \dots, V_n$.

[0040] In an example, the processor 13 may perform the deduplication process with different values $V_1, V_2, \dots, V_n$ of the modifiable parameter 14 in a chronological order or in a random order. A chronological order means that the values $V_1, V_2, \dots, V_n$ are processed in an ascending or a descending order, or in a certain pattern, starting from any desired value. For example, the process may start in a medium range. If the values range from V_1 to V_{10} , then the process may start from V_5 .

[0041] In one example, using a number line, the process may start from V_5 and may use, in a next step, a left-side value, e.g. V_4 and in the following step a right-side value, e.g. V_6 . The processor 13 may compare the results R_4 and R_6 with each other and with the desired result R_D . It may be possible that better results, i.e. results that are closer to the desired result R_D , may be achieved when using left-side values rather than using right-side values. If the processor 13 may detect such a circumstance, the processor 13 may continue to perform the deduplication process using these left-side values. The same would be true, vice versa, for right-side values.

[0042] In an example, the processor 13 may perform the deduplication process with different values $V_1, V_2, \dots, V_n$ in a random order. In this case, the processor 13 may randomly choose any values from the available number of values $V_1, V_2, \dots, V_n$ that may be provided by a parameter 14. This could be used, for instance, if the deduplication process comprises a large number of parameters comprising a large number of values, which would lead to a large number of data and, thus, to a large number of instantiations of the deduplication process. By randomly selecting only some values from the available number of values $V_1, V_2, \dots, V_n$ the processing time could be reduced compared to the time needed for processing all of the available values $V_1, V_2, \dots, V_n$.

[0043] As shown in Figure 2, an example of a computing device 20 comprises a memory 22. The computing device 20 is to store the selected value V_{SEL} of the modifiable parameter 24 associated with a content type identifier CT_x of the input data 25, 26, 27 in the memory 22.

[0044] As described above with reference to Figure 1, the processor 13, 23 may compare the results $R_1, R_2, \dots, R_n$ obtained by performing the deduplication process using respective values $V_1, V_2, \dots, V_n$ with a desired result R_D . The processor 23 may select the value V_{SEL} with which the result closest to the desired result R_D was achieved. This

selected value V_{SEL} may be stored in the memory 22. Thus, the selected value V_{SEL} can be used at a later time at the same site or at a different site.

[0045] In the example shown in Figure 2, the computing device 20 receives input data 25, 26, 27 at the input 21. The input data may, in this example, be content specific. For example, input data 25 may comprise a different content than input data 26. Input data 27 may comprise a content that is different to the content of input data 25 and 26.

[0046] The input data 25, 26, 27 may comprise a label 29A, 29B, 29C with which the content of the respective input data 25, 26, 27 can be identified. For example, input data 25 may comprise a first content CT_1 that may be associated with a first label 29A. Input data 26 may comprise a second content CT_2 that may be associated with a second label 29B. Input data 27 may comprise a third content CT_3 that may be associated with a third label 29C.

[0047] Each input data 25, 26, 27 comprising a respective content CT_1 , CT_2 , CT_3 may be processed by the processor 23. However, depending on the content, the values V_1 , V_2 , ..., V_n of the parameter 24 may yield different results. For example, when input data 25 comprising content CT_1 is processed, then value V_1 may yield the result closest to the desired result R_D . However, when input data 26 comprising a different content, namely content CT_2 is processed, then value V_2 may yield the result that is closest to the desired result R_D . Accordingly, the process is performed content-specifically.

[0048] As shown in Figure 2, based on the content of the input data 25, 26, 27, a respective value V_{SEL} is selected which value yields a result that is closest to the desired result R_D . This selected value V_{SEL} may be stored in the memory 22.

[0049] In an example, the selected value V_{SEL} is stored in the memory 22 together with a content type identifier 29x. The content type identifier 29x is associated with the content of the respective input data that has been processed and for which the value V_{SEL} has been selected. For example, if input data 26 is processed, then value V_{SEL} may be selected as the value which yields a result that is closest to the desired result R_D . As input data 26 comprises content CT_2 , which is represented by label 29B, a respective content type identifier 29x would be stored in the memory 22 together with the selected value V_{SEL} . In this specific case, content type identifier 29x would be associated with content type CT_2 .

[0050] As the content type identifier 29x is stored together with the selected value V_{SEL} , the content type identifier 29x allows to content-specifically identify the value V_{SEL} that yields a result that is closest to the desired result R_D , depending on the respective content CT_X of the respective input data that has been processed.

[0051] In an example, the processor 23 is to determine a content type of other input data 28. Other input data 28 may be input data that is received, for example, at a later point in time or at a different site compared to previously processed input data 26, such as input data 26 mentioned above.

[0052] In the example shown in Figure 2, other input data 28 comprises a content label 29B that is associated with content type CT_2 . The processor 23 is to compare the content type CT_2 of the other input data 28 to the content type CT_X stored in the content type identifier 29x. The processor 23 is to select the stored parameter value V_{SEL} for processing the other input data 28 if the content type CT_2 of the other input data 28 corresponds to the content type CT_X stored in the content type identifier 29x that is associated with the value V_{SEL} . In other words, value V_{SEL} is selected in this example if $CT_X = CT_2$.

[0053] Stated differently, it may be possible that, at a later point in time or at a different site, other input data 28 may be received at the input 21. This other input data 28 may comprise the same content CT_2 as a previously processed input data, such as previously processed input data 26. Thus, a content type identifier 29x associated with this content CT_2 may already be available in the memory 22 as it was stored in the memory when input data 26 was processed. Accordingly, the previously selected value V_{SEL} that is stored together with this content type identifier 29x may now be used by the processor 23 to perform the deduplication process when processing other input data 28.

[0054] Stated yet differently, if input data 26 comprising content type CT_2 is processed, then content type identifier 29x may be associated with CT_2 and may be stored together with the selected value V_{SEL} in the memory 22. At a later point in time, other input data 28 is received at the input 21. In this example, the other input data 28 comprises the same content CT_2 as previously processed input data 26. The other input data 28 and previously processed input data 26 may therefore carry the same content type label 29B.

[0055] The processor 23 may now determine the content type CT_2 of the other input data 28, for instance, by reading the content type label 29B of the other input data

28. The processor 23 may read the memory 22 so as to check whether any values are available in the memory which values may be associated with content type CT_2 . Values that may be available at the memory 22 may be values that have been previously processed, such as, for example, value V_{SEL} that has been selected previously when input data 26 was processed, as discussed above. Additionally or alternatively, values that may be available at the memory 22 may also be values that have been previously stored, for example as a preset of values, by a user.

[0056] Still with reference to the aforementioned example shown in Figure 2, if the processor 23 determines that a value V_{SEL} is available in the memory 22 which value V_{SEL} is associated with content type CT_2 , then this value V_{SEL} is selected for processing the other input data 28. Accordingly, a simulation of the deduplication process using one or more of the entire range of available values $V_1, V_2, \dots, V_n$ may be omitted because the value V_{SEL} that yields a result closest to the desired result R_D is already available in the memory 22.

[0057] In an example shown in Figure 3, the deduplication process comprises a chunker process 30 to split the input data 35 into chunks 36A, 36B, 36C of variable size, wherein the modifiable parameter 34 represents a minimum chunk size 34A or a maximum chunk size 34B of a chunk.

[0058] The input data 35 is processed in the deduplication process, wherein the deduplication process comprises the chunker process 30. The chunker process 30 may be a part of the deduplication process.

[0059] In Figure 3, there are two options available for processing the input data 35. The first option, as indicated by the branch 31A, is to divide the input data 35 into chunks 35A of a minimum size. After a value V_{SEL} for the minimum chunk size has been selected, as indicated by branch 32A, the input data is divided into chunks 36A, 36B, 36C comprising this minimum chunk size.

[0060] The second option, as indicated by the branch 31B, is to divide the input data 35 into chunks 35B of a maximum size. After a value V_{SEL} for the maximum chunk size has been selected, as indicated by branch 32B, the input data is divided into chunks 36A, 36B, 36C comprising this maximum chunk size.

[0061] Regarding the first option, the input data 35A is processed by the chunker process 30. The modifiable parameter 34A represents a minimum chunk size, wherein the

values $V_1, \dots, V_m$ represent a range of values within the minimum chunk size. A chunk size may, for instance, be represented by a number of Bits, Nibbles or Bytes. For example, a minimum chunk size may vary between a value of 3 Bits and a value of 6 Bits. Depending on the selected value V_{SEL} of the minimum chunk size, the input data 35 may be divided into chunks 36A, 36B, 36C comprising a chunk size corresponding to the selected number of Bits V_{SEL} .

[0062] Regarding the second option, the input data 35B is processed by the chunker process 30. The modifiable parameter 34B represents a maximum chunk size, wherein the values $V_{m+1}, \dots, V_n$ represent a range of values within the maximum chunk size. A chunk size may, for instance, be represented by a number of Bits, Nibbles or Bytes. For example, a maximum chunk size may vary between a value of 7 Bits and a value of 10 Bits. A maximum chunk size is greater than a minimum chunk size, wherein the lowest value V_{m+1} of the range of the maximum chunk size is greater, by at least one unit, than the greatest value V_m of the range of the minimum chunk size.

[0063] Depending on the selected value V_{SEL} of the maximum chunk size, the input data 35 may be divided into chunks 36A, 36B, 36C comprising a chunk size corresponding to the selected number of Bits V_{SEL} .

[0064] In another example, the modifiable parameter is a first modifiable parameter 34A representing a minimum chunk size and the deduplication process comprises a second modifiable parameter 34B representing a maximum chunk size. While the functional principle of this example may be similar to the one described above, with reference to Figure 3, the difference to the above mentioned example is that the deduplication process comprises two modifiable parameters 34A, 34B. Thus, both the minimum chunk size $V_1, \dots, V_m$ and the maximum chunk size $V_{m+1}, \dots, V_n$ can be selected together.

[0065] In another example, as shown in Figure 4, the deduplication process comprises a plurality of modifiable parameters 41, 42, 43, wherein the processing result $R_{1n}, R_{2n}, \dots, R_{xn}$ depends on the value $V_{1n}, V_{2n}, \dots, V_{xn}$ of each of the modifiable parameters 41, 42, 43, wherein the processor is to perform the process a plurality of times each time with a different value of each of the modifiable parameters 41, 42, 43 and to obtain a result for each value of each modifiable parameter 41, 42, 43, and wherein the processor is to select a value of each of the modifiable parameters 41, 42, 43 associated

with a result closest to a desired result as the value of each of the modifiable parameters 41, 42, 43 to be used in processing the input data.

[0066] For example, the deduplication process may comprise a first modifiable parameter 41, a second modifiable parameter 42 and a x^{th} modifiable parameter 43 with one or more modifiable parameters in between.

[0067] The first modifiable parameter 41 may comprise several values V_{11} , V_{12} , ..., V_{1n} . Depending on the value V_{11} , V_{12} , ..., V_{1n} that has been selected for performing the deduplication process, a respective result R_{11} , R_{12} , ..., R_{1n} may be obtained.

[0068] For example, as shown in Figure 4, if the value V_{11} is selected for performing the deduplication process, the respective result that may be obtained is result R_{11} . If the value V_{12} is selected for performing the deduplication process, the respective result that may be obtained is result R_{12} . Stated in more general terms, if the value V_{1n} is selected for performing the deduplication process, the respective result that may be obtained is result R_{1n} , wherein the letter 'n' is a natural number and represents the index of a plurality of values and results, respectively.

[0069] The same applies correspondingly to the second modifiable parameter 42 and to the x^{th} modifiable parameter, as well as to any modifiable parameters in between.

[0070] In an example, the deduplication process comprises a plurality of modifiable parameters 41, 42, 43. According to the example shown in Figure 4, the deduplication process comprises at least two modifiable parameters 41, 42, which represent a plurality of modifiable parameters. Accordingly, not all of the modifiable parameters described with reference to and shown in Figure 4 are required in all of the examples.

[0071] The processor is to perform the deduplication process a plurality of times, each time with a different value of each of the modifiable parameters 41, 42, 43. Accordingly, the processor is to perform a simulation of the deduplication process. The processor may obtain a result R_{xn} for each value V_{xn} of each modifiable parameter 41, 42, 43. The processor may compare the results with each other. The processor may also compare the results with a desired result R_D .

[0072] For example, the processor may determine that a result closest to a desired result R_D may be achieved when the first parameter 41 uses the value V_{1n} and when the

second parameter uses the value V_{2n} . Thus, the processor may select these values V_{1n} , V_{2n} for processing the input data in order to obtain a result that is closest to the desired result R_D .

[0073] In order to determine the respective values, the processor may, for instance, perform the deduplication process, or the simulation of the deduplication process, respectively, with different combinations of the values of the first modifiable parameter 41 and the second modifiable parameter 42. The values may be combined such that all possible combinations and/or variations are covered.

[0074] Still with respect to Figure 4, it may be possible that each combination of different values of different parameters may yield different results R_{1D} , R_{2D} , R_{nD} . For example, a combination of the value V_{11} with the value V_{21} may yield a result R_{1D} . A combination of the value V_{12} with the value V_{22} may yield a result R_{2D} . Stated more generally, a combination of a first value V_{1n} with a second value V_{2n} may yield a result R_{nD} .

[0075] The processor may compare the results R_{1D} , R_{2D} , R_{nD} with the desired result R_D . The processor may select that combination of values that yields the result that is closest to the desired result R_D . The processor may perform the deduplication process with the selected combination of values.

[0076] As an example, the result R_{2D} may be the result that is closest to the desired result R_D . The result R_{2D} is obtained by using the value V_{12} for the first parameter 41 and the value V_{22} for the second parameter 42. Accordingly, the processor may use the value V_{12} for the first parameter 41 and the value V_{22} for the second parameter 42 to process the input data.

[0077] The processor may perform a simulation of the deduplication process before the processor processes the input data. In this simulation, different combinations of values of the modifiable parameters may be used to obtain a respective result, as described above. The processor may compare the respective result R_{1D} , R_{2D} , ..., R_{nD} with the desired result R_D . The processor may determine the result that is closest to the desired result R_D . The processor may select those values, i.e. that combination of values with which the result closest to the desired result R_D was achieved. The processor may use these selected values to process the input data, i.e. to perform the deduplication process as part of an actual deduplication performed subsequent to the performance of the simulation of the deduplication process.

[0078] The selected values may be stored in a memory for later use, for instance, at the same site or at a different site.

[0079] As described above with reference to Figure 2, the selected values may be associated with a content type identifier in order to allow the processor to identify the content of other input data 28 that may, for instance, arrive later in time or at a different site.

[0080] Such an example is shown in Figure 5. The difference to the example of Figure 2 is that the deduplication process may comprise a plurality of parameters, for example, a first parameter 41 and a second parameter 42.

[0081] The selected value V_{SEL1} of the first parameter 41 and the selected value V_{SEL2} of the second parameter 42 may be stored in the memory 22. As described above, the values V_{SEL1} and V_{SEL2} may have been selected by the processor 23 as they provide a result that is closest to a desired result. For example, the selected value V_{SEL1} of the first parameter 41 may correspond to the value V_{12} and the selected value V_{SEL2} of the second parameter 42 may correspond to the value V_{22} , wherein the result obtained by using these two parameter values V_{12} and V_{22} may be result R_{2D} . Result R_{2D} may be a result that is closest to a desired result R_D .

[0082] The selected values V_{SEL1} and V_{SEL2} may be stored in the memory 22 together with a content type identifier 29x. As described above with reference to Figure 2, the content type identifier 29x may be associated with the content type of the input data that has been processed. For example, if the input data 26, which comprises content CT_2 , was processed and the values V_{SEL1} and V_{SEL2} have been selected by the processor for processing input data 26, then the content type identifier 29x may be associated with content type CT_2 .

[0083] If other input data 28 is processed, which other input data 28 may comprise the same content type CT_2 that could be indicated by label 29B, then the processor 23 may check whether values, for example V_{SEL1} and V_{SEL2} , may be available in the storage 22, which values may be associated with content type CT_2 , for example by content type identifier 29x. If so, the processor 23 may use these values V_{SEL1} and V_{SEL2} for processing the other input data 28.

[0084] Briefly summarizing, the processor is to process input data of a certain content type by performing a deduplication process. In order to achieve the best

deduplication result for this input data, a simulation of the deduplication process is performed prior to the execution of the actual deduplication process. However, such a simulation may be omitted if the memory already contains a value that may lead to the best result. Such a value may be available in the memory if input data containing similar or the same content may have already been processed earlier, and wherein the simulation yielded said value. In other words, it may be sufficient if a simulation is performed when input data containing a certain content type is processed for the first time.

[0085] Figure 6 shows a block diagram that depicts an example of a method for processing input data, which method is implemented by a computing device. The order of the blocks shown in Figure 6 may not be identical to the order in which the method steps may be performed.

[0086] At block 61 input data is received by the computing device.

[0087] Block 62 indicates that a deduplication process is to be performed by the computing device to process the input data. The deduplication process comprises a modifiable parameter and the processing result depends on the value of the modifiable parameter.

[0088] As indicated by block 63, a simulation may be performed prior to performing the actual deduplication process of block 62. The simulation may be performed in that the deduplication process is performed, by the computing device, a plurality of times, each time with a different value of the modifiable parameter. A respective result is obtained for each value.

[0089] At block 64 the value associated with a result closest to a desired result is selected by the computing device as the value to be used in processing the input data in the deduplication process that is to be performed subsequent to the aforementioned simulation of the deduplication process.

[0090] At block 65 the selected value of the modifiable parameter, which value may have been selected during the simulation, is stored in a memory together with a content type identifier.

[0091] At block 66 the content type of other input data is compared to the previously stored content type identifier. Other input data may be, for instance, input data

that may arrive later in time or at a different site compared to the input data previously processed during the simulation.

[0092] At block 67 the stored parameter value is selected for processing the other input data, if the content type of the other input data corresponds to the stored content type identifier.

[0093] Figure 7 shows a flow chart of an example of such a method.

[0094] At block 71 input data is received. At block 72 the content type of the respective input data is determined. This could, for example, be achieved in that a content type identifier or a content type label 29B, that may be carried with the received input data 26, is read. For example, it is determined that the input data 26 may comprise a content type CT_2 .

[0095] At block 73 it is checked whether there might be any value V_{SEL} available that is associated with that content type CT_2 . For example, the value V_{SEL} may be available in the memory 22. The value V_{SEL} may comprise a content type identifier 29x that is associated with content type CT_2 . The value V_{SEL} may have been stored during a previous performance of the deduplication process or of a simulation of the deduplication process, or it may have been stored previously as a preset, for example, by a human user.

[0096] If a value V_{SEL} associated with this content type CT_2 should be available, as indicated by branch 74, the stored value V_{SEL} is used for processing the input data 26, as indicated by block 75.

[0097] If a value V_{SEL} associated with this content type CT_2 should not be available, as indicated by branch 76, a respective value is determined, as indicated by block 77. The determination of the value may, for instance, be achieved by performing a simulation of the deduplication process and selecting the value associated with a result closest to a desired result.

[0098] If a value is selected, this selected value V_{SEL} may be stored in a memory 22 together with a content type identifier 29x that represents the content type CT_2 of the input data 26, as indicated by block 78.

[0099] As shown in Figure 8, examples 80 relate to a non-transitory machine-readable storage medium 81 encoded with instructions executable by a processing

resource 82 of a computing device 83 to perform methods described herein, in particular to cause the processing resource 82 to perform a method to perform a deduplication process to process input data, wherein the deduplication process comprises a modifiable parameter and wherein a processing result depends on the value of the modifiable parameter.

[00100] The method is further to perform a simulation of the deduplication process, wherein the deduplication process is performed a plurality of times, each time with a different value of the modifiable parameter, and to obtain a result for each value.

[00101] The method is further to select the value associated with a result closest to a desired result as the value to be used in processing the input data.

[00102] The method is further to store the selected value of the modifiable parameter associated with a content type identifier in memory 81.

[00103] The method is further to compare the content type of other input data to the stored content type identifier.

[00104] The method is further to select the stored parameter value for processing the other input data if the content type of the other input data corresponds to the stored content type identifier.

[00105] It would be appreciated that examples described herein can be realized in the form of hardware, machine readable instructions or a combination of hardware and machine readable instructions. Any such machine readable instructions may be stored in the form of volatile or non-volatile storage such as, for example, a storage device such as a ROM, whether erasable or rewritable or not, or in the form of memory such as, for example, RAM, memory chips device or integrated circuits or an optically or magnetically readable medium such as, for example, a CD, DVD, magnetic disc or magnetic tape. It would be appreciated that these storage devices and storage media are examples of machine readable storage that are suitable for storing a program or programs that, when executed, implement examples described herein.

[00106] All of the features disclosed in the specification (including any accompanying claims, abstract and drawings) and/or all of the features of any method or process disclosed may be combined in any combination, except combinations where at least some of such features are mutually exclusive.

[00107] Each feature disclosed in the specification (including any accompanying claims, abstract and drawings) may be replaced by alternative features serving the same, equivalent or similarly proposed, unless expressly stated otherwise. Thus, unless expressly stated otherwise, each feature disclosed is one example only of a generic series of equivalent or similar features.

Claims

We claim:

1. A computing device comprising:
 - an input to receive input data;
 - a processor to perform a deduplication process to process the input data, wherein the deduplication process comprises a modifiable parameter and wherein a processing result depends on the value of the modifiable parameter;
 - wherein the processor is to perform a simulation of the deduplication process, wherein the deduplication process is performed a plurality of times, each time with a different value of the modifiable parameter, and to obtain a result for each value;
 - wherein the processor is to select the value associated with a result closest to a desired result as the value to be used in processing the input data; and
 - wherein the processor is to store the selected value of the modifiable parameter associated with a content type identifier in memory, compare the content type of other input data to the stored content type identifier, and to select the stored parameter value for processing the other input data if the content type of the other input data corresponds to the stored content type identifier.
2. The computing device of claim 1, wherein the modifiable parameter has a number of possible values and wherein the processor is to perform the deduplication process with each one of the possible values.
3. The computing device of claim 2, wherein the processor is to perform the deduplication process with different values of the modifiable parameter in a chronological order or in a random order.
4. The computing device of claim 1, wherein the deduplication process comprises a chunker process to split the input data into chunks of variable size, wherein the modifiable parameter represents a minimum chunk size or a maximum chunk size of a chunk.

5. The computing device of claim 4, wherein the modifiable parameter is a first modifiable parameter representing a minimum chunk size and wherein the deduplication process comprises a second modifiable parameter representing a maximum chunk size.
6. The computing device of claim 1, wherein the deduplication process comprises a plurality of modifiable parameters, wherein the processing result depends on the value of each of the modifiable parameters, wherein the processor is to perform the process a plurality of times each time with a different value of each of the modifiable parameters and to obtain a result for each value of each modifiable parameter, and wherein the processor is to select a value of each of the modifiable parameters associated with a result closest to a desired result as the value of each of the modifiable parameters to be used in processing the input data.
7. A method implemented by a computing device, the method comprising:

receiving input data by the computing device;

performing, by the computing device, a deduplication process to process the input data, wherein the deduplication process comprises a modifiable parameter and wherein a processing result depends on the value of the modifiable parameter;

performing, by the computing device, a simulation of the deduplication process by performing the deduplication process a plurality of times each time with a different value of the modifiable parameter and obtaining a result for each value;

selecting, by the computing device, the value associated with a result closest to a desired result as the value to be used in processing the input data in the deduplication process to be performed subsequent to the simulation of the deduplication process;

storing the selected value of the modifiable parameter associated with a content type identifier in memory;

comparing the content type of other input data to the stored content type identifier; and

selecting the stored parameter value for processing the other input data if the content type of the other input data corresponds to the stored content type identifier.

8. The method of claim 7, wherein the modifiable parameter has a number of possible values and wherein the method comprises performing the deduplication process with each one of the possible values.
9. The method of claim 8, wherein the method comprises performing the deduplication process with different values of the modifiable parameter in a chronological order or in a random order.
10. The method of claim 7, wherein the process comprises a plurality of modifiable parameters, wherein the processing result depends on the value of each of the modifiable parameters, wherein the method comprises performing the process a plurality of times each time with a different value of each of the modifiable parameters and obtaining a result for each value of each modifiable parameter, and selecting a value of each of the modifiable parameters associated with a result closest to a desired result as the value of each of the modifiable parameters to be used in processing the input data.
11. A non-transitory machine-readable storage medium encoded with instructions executable by a processing resource of a computing device to cause the processing resource to perform a method to:

perform a deduplication process to process input data, wherein the deduplication process comprises a modifiable parameter and wherein a processing result depends on the value of the modifiable parameter;

perform a simulation of the deduplication process, wherein the deduplication process is performed a plurality of times, each time with a different value of the modifiable parameter, and to obtain a result for each value;

select the value associated with a result closest to a desired result as the value to be used in processing the input data;

store the selected value of the modifiable parameter associated with a content type identifier in memory;

compare the content type of other input data to the stored content type identifier; and

select the stored parameter value for processing the other input data if the content type of the other input data corresponds to the stored content type identifier.

12. The non-transitory machine-readable storage medium of claim 11, wherein the modifiable parameter has a number of possible values and wherein the processing resource is to perform the deduplication process with each one of the possible values.
13. The non-transitory machine-readable storage medium of claim 11, wherein the processing resource is to perform the deduplication process with different values of the modifiable parameter in a chronological order or in a random order.
14. The non-transitory machine-readable storage medium of claim 11, wherein the deduplication process comprises a chunker process to split the input data into chunks of variable size, wherein the modifiable parameter represents a minimum chunk size or a maximum chunk size of a chunk.
15. The non-transitory machine-readable storage medium of claim 11, wherein the deduplication process comprises a plurality of modifiable parameters, wherein the processing result depends on the value of each of the modifiable parameters, wherein the processor is to perform the process a plurality of times each time with a different value of each of the modifiable parameters and to obtain a result for each value of each modifiable parameter, and wherein the processor is to select a value of each of the modifiable parameters associated with a result closest to a desired result as the value of each of the modifiable parameters to be used in processing the input data.

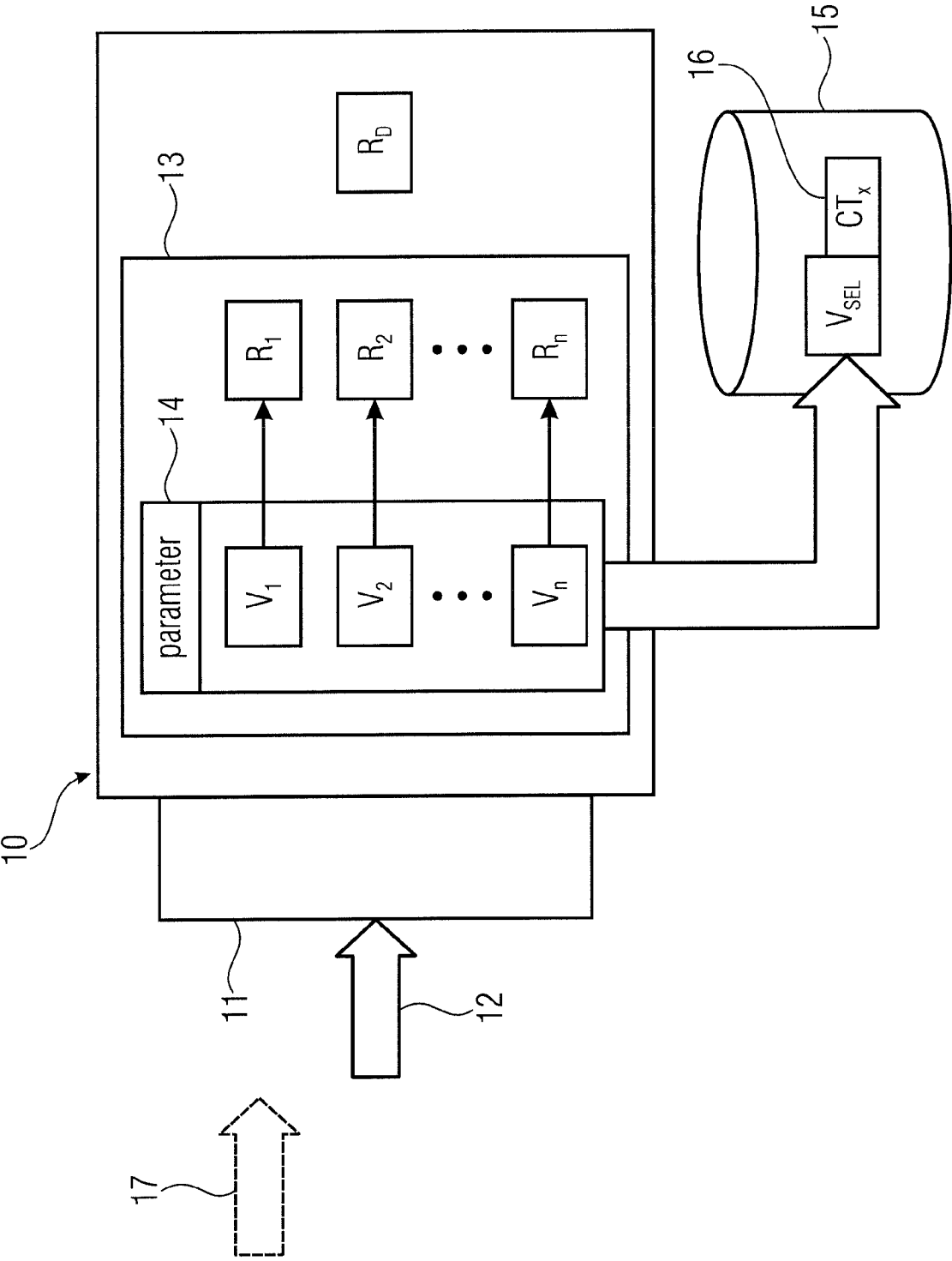


FIG 1

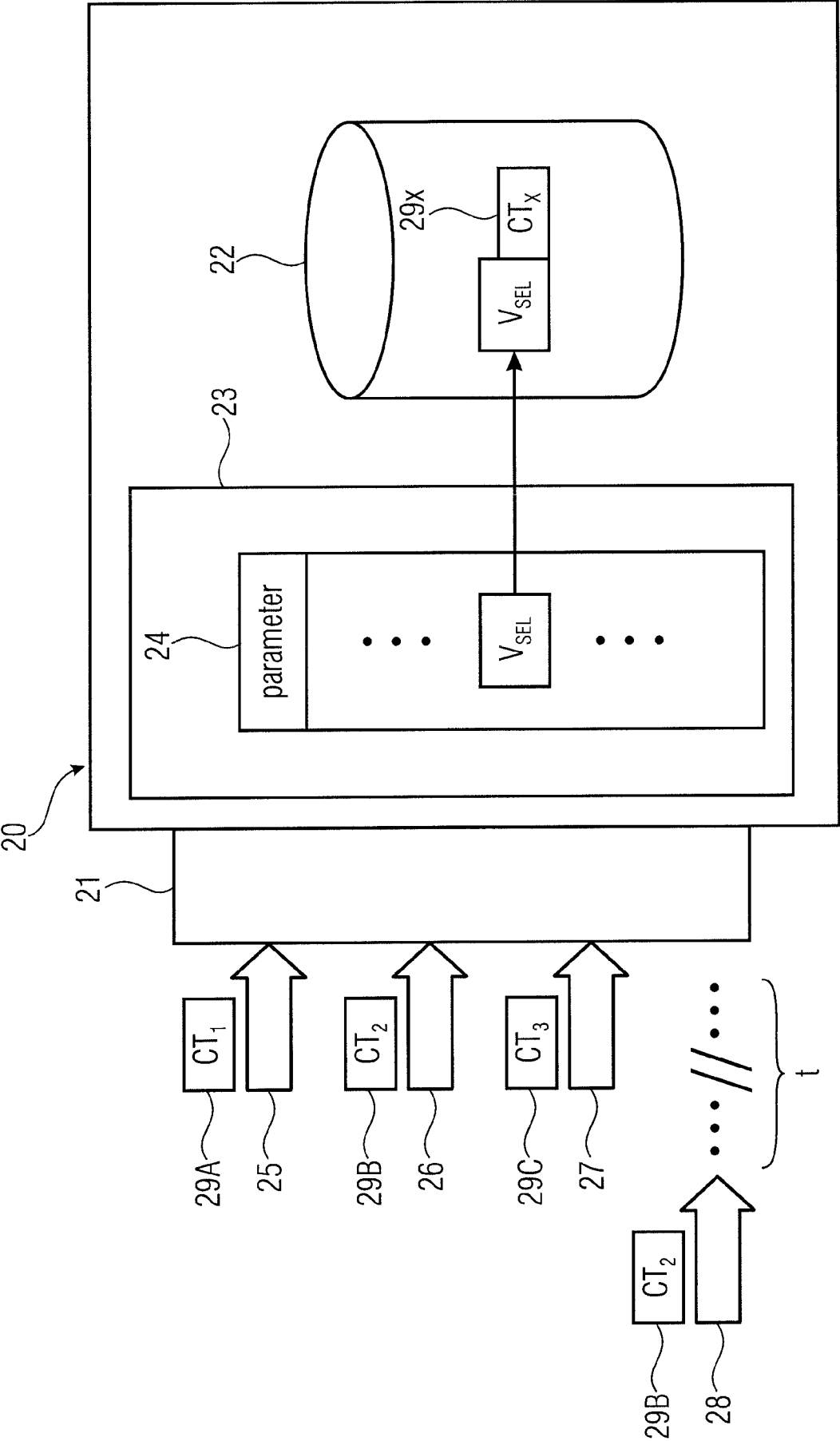


FIG 2

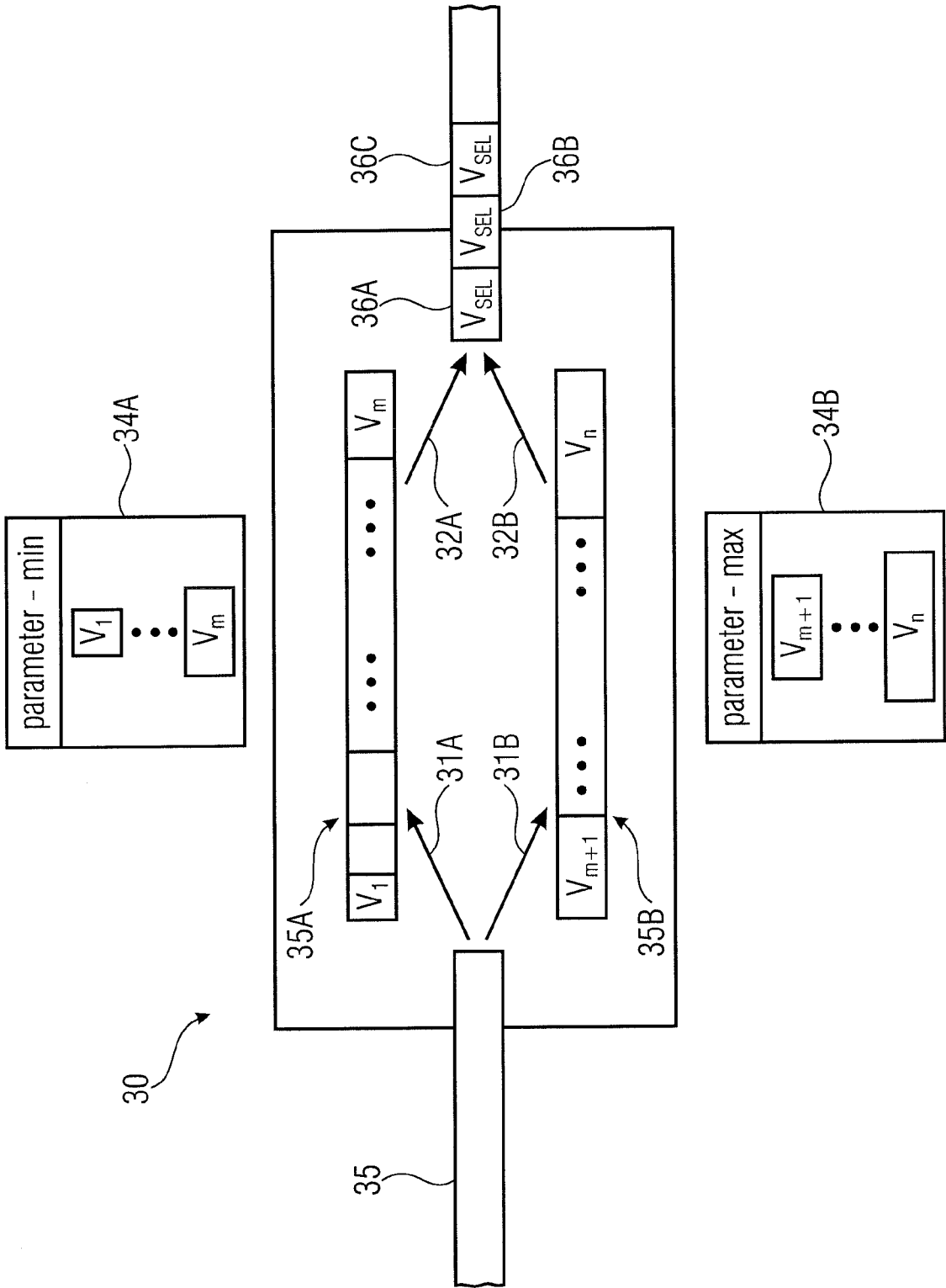


FIG 3

4/8

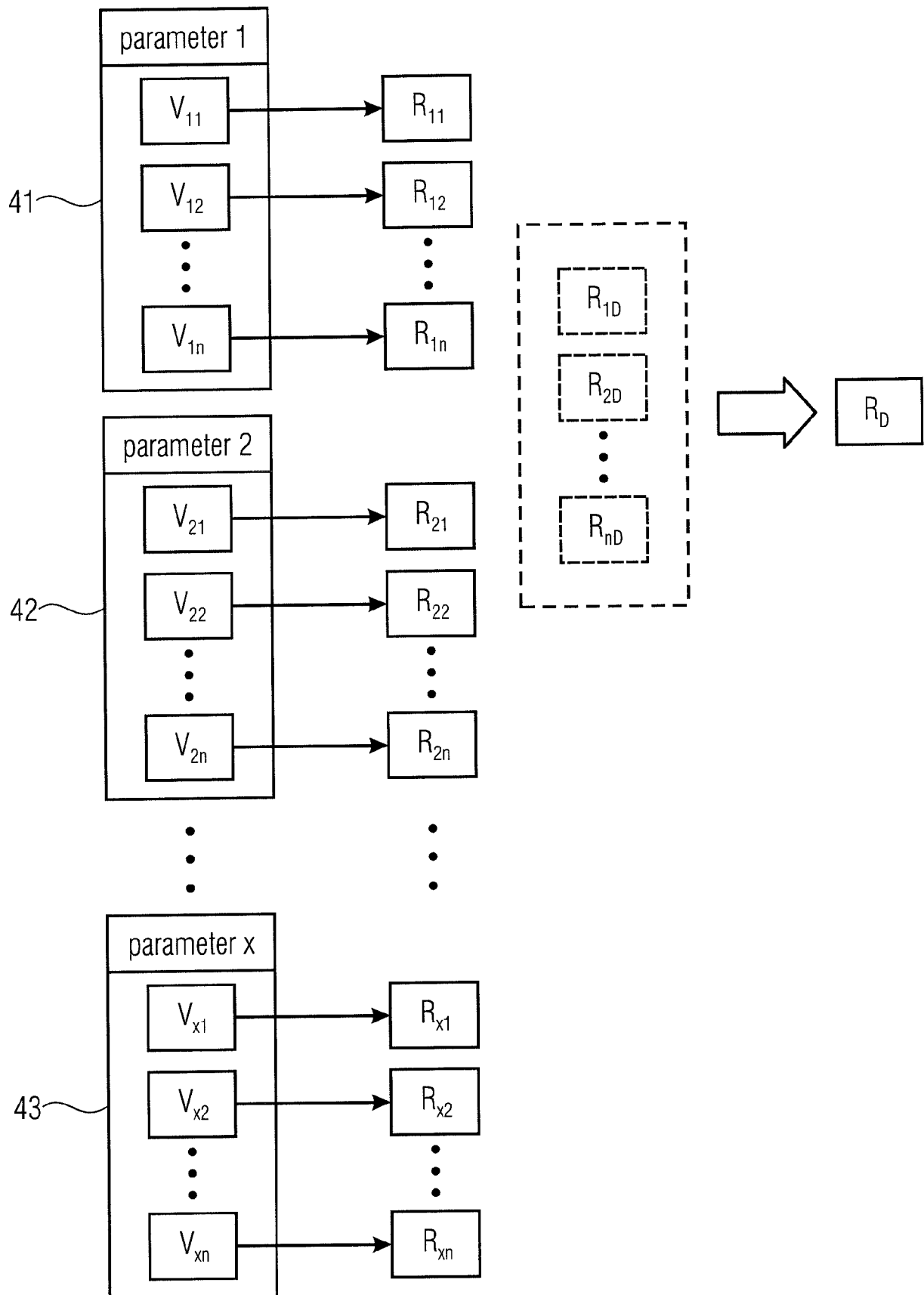


FIG 4

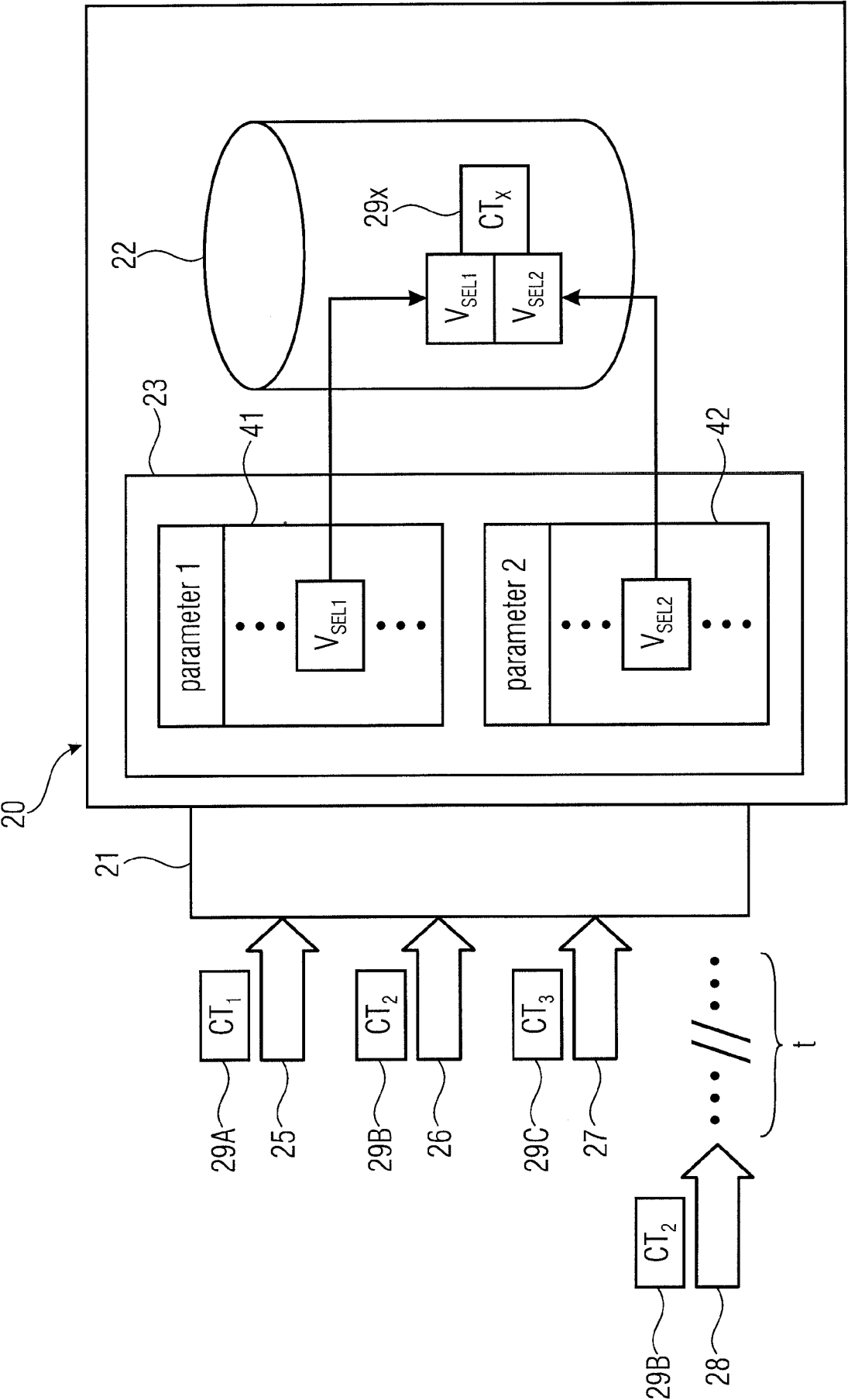


FIG 5

6/8

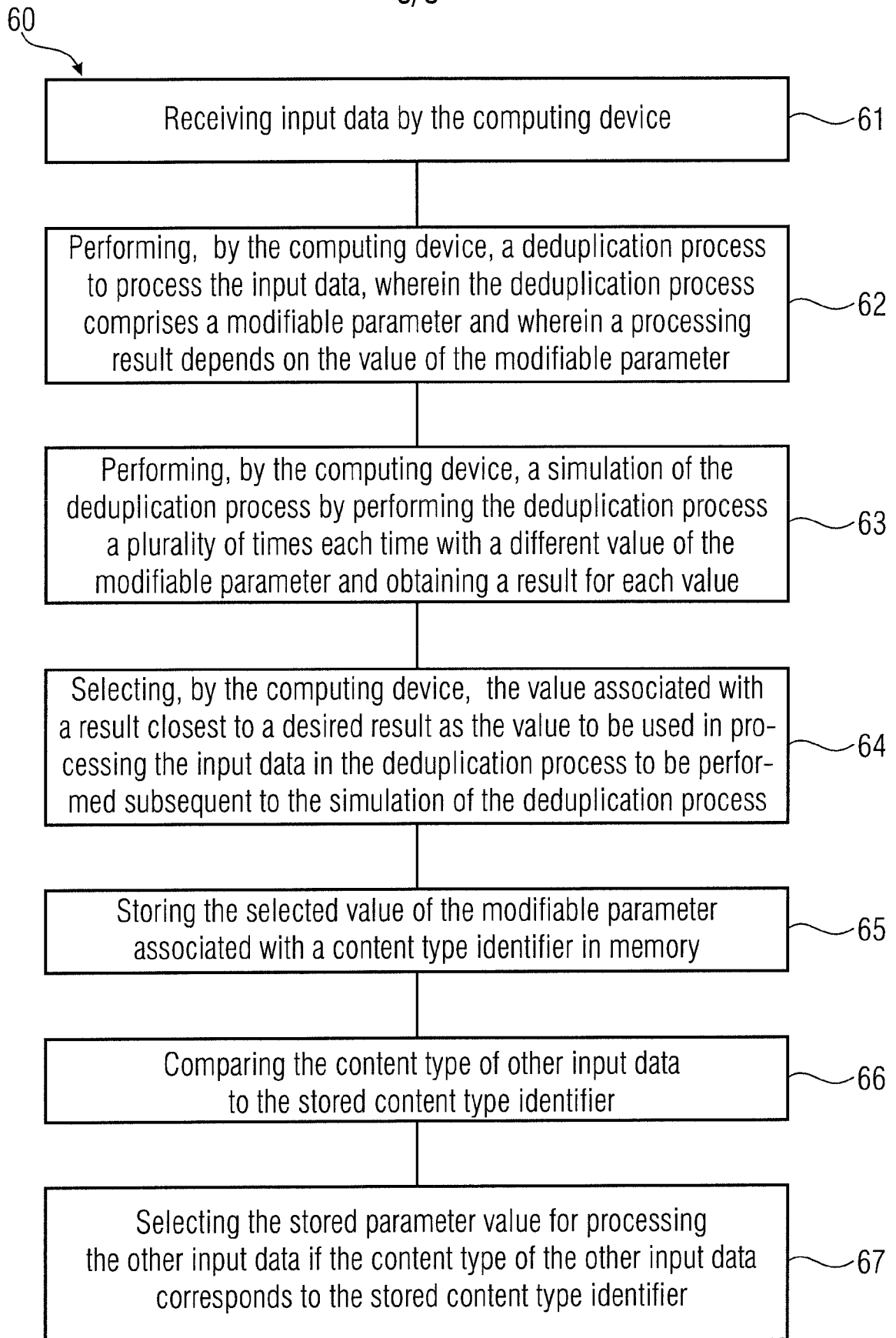


FIG 6

7/8

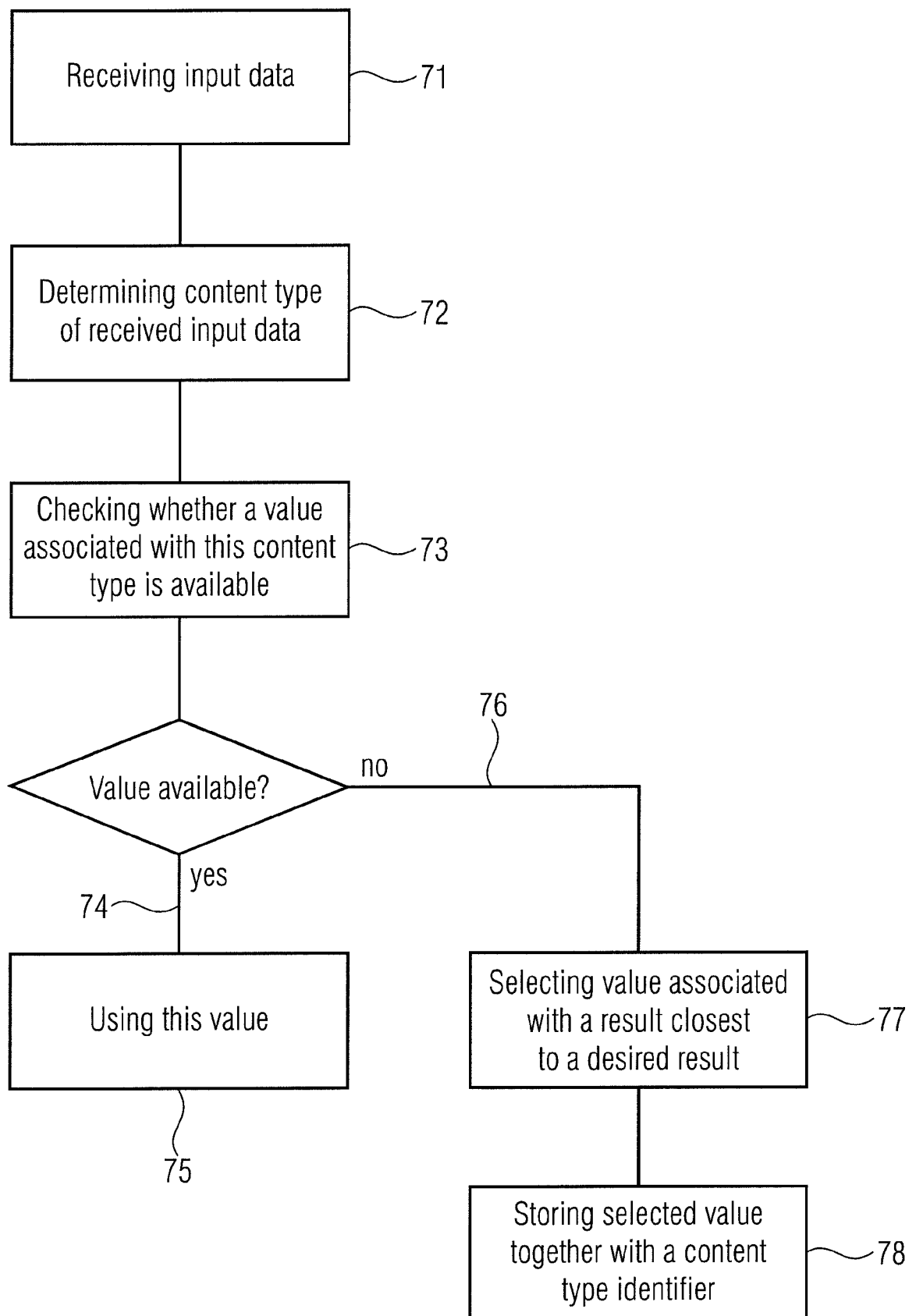


FIG 7

8/8

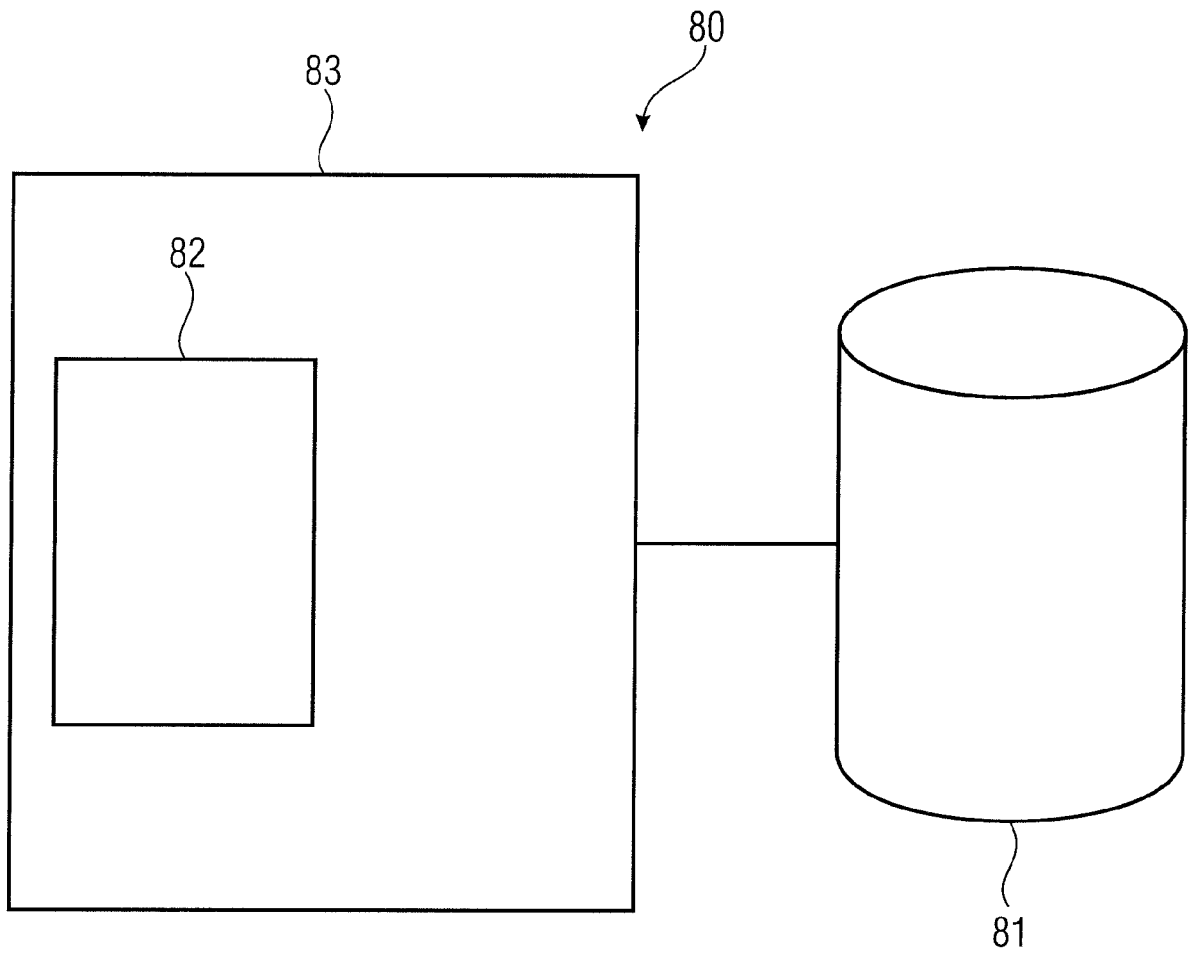


FIG 8

INTERNATIONAL SEARCH REPORT

International application No
PCT/EP2015/059394

A. CLASSIFICATION OF SUBJECT MATTER

INV. G06F17/30 G06F3/06 G06F11/14 G06F9/445
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, WPI Data

C. DOCUMENTS CONSIDERED TO BE RELEVANT

Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
X	US 2014/095439 A1 (RAM TAMIR [US]) 3 April 2014 (2014-04-03) figures 1-2 claims 1-12 paragraph [0007] - paragraph [0062] -----	1-15
A	US 2010/198797 A1 (WIDEMAN RODERICK B [US]) 5 August 2010 (2010-08-05) the whole document -----	1-15
A	US 2011/125719 A1 (JAYARAMAN VINOD [US]) 26 May 2011 (2011-05-26) the whole document -----	1-15



Further documents are listed in the continuation of Box C.



See patent family annex.

* Special categories of cited documents :

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

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

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

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

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

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

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

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

"&" document member of the same patent family

Date of the actual completion of the international search

7 January 2016

Date of mailing of the international search report

15/01/2016

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

Rackl, Günther

INTERNATIONAL SEARCH REPORT

Information on patent family members

International application No

PCT/EP2015/059394

Patent document cited in search report	Publication date	Patent family member(s)	Publication date
US 2014095439 A1	03-04-2014	CN 104871155 A	26-08-2015
		EP 2904525 A1	12-08-2015
		US 2014095439 A1	03-04-2014
		WO 2014055449 A1	10-04-2014

US 2010198797 A1	05-08-2010	GB 2472963 A	23-02-2011
		US 2010198797 A1	05-08-2010
		WO 2010090970 A2	12-08-2010

US 2011125719 A1	26-05-2011	US 2011125719 A1	26-05-2011
		US 2013159262 A1	20-06-2013
