

(12) 特許協力条約に基づいて公開された国際出願

(19) 世界知的所有権機関
国際事務局

(43) 国際公開日
2018年4月26日(26.04.2018)



(10) 国際公開番号

WO 2018/073940 A1

(51) 国際特許分類:
G06F 12/00 (2006.01)

(21) 国際出願番号: PCT/JP2016/081142

(22) 国際出願日: 2016年10月20日(20.10.2016)

(25) 国際出願の言語: 日本語

(26) 国際公開の言語: 日本語

(71) 出願人: 楽天株式会社 (RAKUTEN, INC.) [JP/JP]; 〒1580094 東京都世田谷区玉川一丁目14番1号 Tokyo (JP).

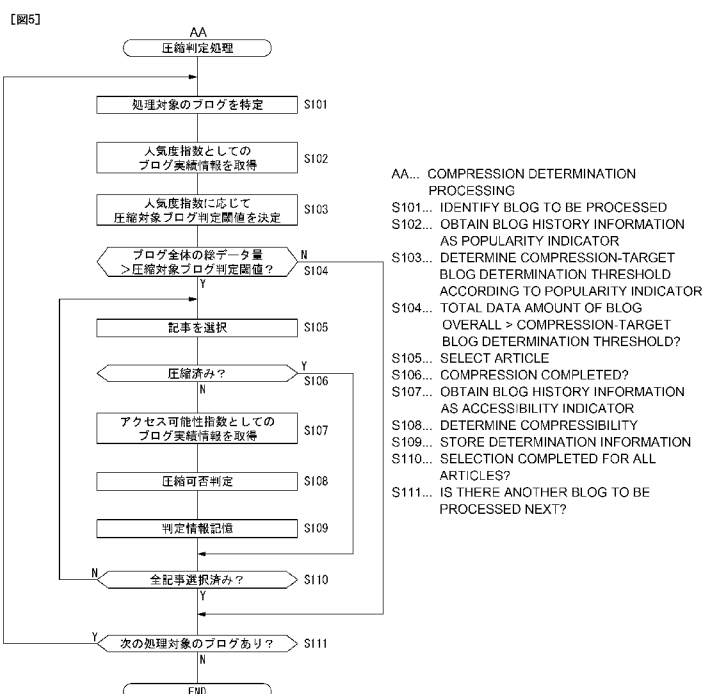
(72) 発明者: 山本 陽司 (YAMAMOTO, Yoji); 〒1580094 東京都世田谷区玉川一丁目14番1号 楽天株式会社内 Tokyo (JP).

(74) 代理人: 岩田 雅信, 外 (IWATA, Masanobu et al.); 〒1010032 東京都千代田区岩本町1丁目3番9号 ハクセイビル8階 テクノピア国際特許事務所 Tokyo (JP).

(81) 指定国(表示のない限り、全ての種類の国内保護が可能): 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, DJ, 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, KW, 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,

(54) Title: INFORMATION PROCESSING DEVICE, INFORMATION PROCESSING METHOD, PROGRAM AND STORAGE MEDIUM

(54) 発明の名称: 情報処理装置、情報処理方法、プログラム、記憶媒体



(57) Abstract: An increase in the size and/or number of blogs in a server for providing a blog service gradually exhausts memory resources. In response, the present invention reduces this memory load. To this end, the present invention is equipped with: a popularity indicator acquisition unit for acquiring a popularity indicator for a blog which contains one or more articles; a threshold setting unit for setting a threshold for determining whether or not to compress each of said blogs according to the popularity indicator of each blog; and a determination unit for determining whether or not a blog

[続葉有]

SY, TH, TJ, TM, TN, TR, TT, TZ, UA, UG, US, UZ,
VC, VN, ZA, ZM, ZW.

- (84) 指定国 (表示のない限り、全ての種類の広域保護が可能): ARIPO (BW, GH, GM, KE, LR, LS, MW, MZ, NA, RW, SD, SL, ST, SZ, TZ, UG, ZM, ZW), ユーラシア (AM, AZ, BY, KG, KZ, RU, TJ, TM), ヨーロッパ (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).

添付公開書類:

- 一 国際調査報告 (条約第21条(3))

should be a compression-target blog, in which at least some of the articles contained in said blog are compressed, on the basis of the threshold and the total amount of data constituting the articles contained in said blog, and also determining compression-target articles, which are the articles to be compressed among the articles contained in the compression-target blog.

(57) 要約: ブログサービスを提供するサーバにおいてブログの数やサイズの増大によりメモリリソースが徐々に圧迫されていくため、このメモリ負担を低減する。そこで1または複数の記事が含まれるブログの人気度指数を取得する人気度指数取得部と、ブログごと的人气度指数に応じて該ブログごとに圧縮するか否かを判定するための閾値を設定する閾値設定部と、一つのブログに含まれる記事の総データ量と閾値とに基づいて当該一つのブログに含まれる記事の少なくとも一部を圧縮する圧縮対象ブログとするか否かを判定し、圧縮対象ブログに含まれる各記事の中から圧縮対象とする記事を圧縮対象記事として判定する判定部と、を備えるものとする。

明 細 書

発明の名称：

情報処理装置、情報処理方法、プログラム、記憶媒体

技術分野

[0001] 本発明は情報処理装置、情報処理方法、プログラム、記憶媒体に関し、特にブログを管理するサーバ装置への適用に好適な技術に関する。

背景技術

[0002] インターネットを利用したサービスの一つとしてブログ環境を提供するサービスが知られている。ユーザはブログとしての任意の文章や画像による記事をサーバにアップロードし、当該ユーザの書き込み記事として保存してもらう。保存されたブログは、例えばウェブページの形式で一般に公開される。また、公開範囲を限定したものや、非公開のものなどがある。

多くのユーザは、自己発信のためのツールや、或いは個人的な日記代わりなどとして、このようなブログサービスを利用している。

特許文献1には、ブログに関する技術、例えば記事のアップロードに関する技術が開示されている。

特許文献2には、サーバの残容量不足に応じたコンテンツ削除に関する技術が開示されている。

先行技術文献

特許文献

[0003] 特許文献1：特開2007-328750号公報

特許文献2：特開2010-44468号公報

発明の概要

発明が解決しようとする課題

[0004] ブログは無料で始められるものも多く、とりあえず始めてみたがほとんど更新しないユーザがいる一方で、頻繁に記事を更新して人気を集めるユーザがいるように、玉石混淆である。従って、ブログや個々の記事におけるアク

セス数はさまざまである。

ブログが開設されてから2、3の記事がアップロードされたものの、その後更新されずにアクセスもないまま長期間放置され、開設したユーザによるログイン操作も行われないようなブログは、当該ユーザにとっても不要である場合がある。このようなブログは、ブログサービスを提供する事業者にとっても徒に記憶リソースを圧迫するものとなる。

[0005] そこで、アクセス可能性の低いブログや記事については圧縮しておき、アクセスがあった場合に解凍して配信することが考えられる。

但し、圧縮や解凍の処理は負荷が高い。アクセスの際に記事の解凍が必要となると、その処理時間により閲覧ユーザにウェブページのレスポンスの遅さを感じさせるなど閲覧時のパフォーマンス低下を生じさせる恐れがある。これらのことから、閲覧時における圧縮や解凍の処理回数は、可能な限り減らしたい。

そこで本発明は、圧縮する記事を適切に選択することにより、閲覧時の解凍処理を可能な限り生じさせないようにしつつ、記憶リソースの有効な利用を図ることを目的とする。

課題を解決するための手段

[0006] 本発明に係る情報処理装置は、1または複数の記事が含まれるブログの人気度指数を取得する人気度指数取得部と、前記ブログごとの前記人気度指数に応じて該ブログごとに圧縮するか否かを判定するための閾値を設定する閾値設定部と、一つの前記ブログに含まれる記事の総データ量と前記閾値とに基づいて当該一つの前記ブログに含まれる記事の少なくとも一部を圧縮する圧縮対象ブログとするか否かを判定し、前記圧縮対象ブログに含まれる各記事の中から圧縮対象とする記事を圧縮対象記事として判定する判定部と、を備えている。

ブログに含まれる記事によって当該ブログを記憶するためのリソースが圧迫された場合に、当該ブログに属する記事を圧縮することを考える。このときに、圧縮するか否かを判定するための閾値をブログごとの人気度指数に基

づいて決める。

[0007] 上記した情報処理装置においては、前記圧縮対象ブログに属する記事のうち前記圧縮対象記事の圧縮を行い、既に圧縮した記事に対するアクセス要求が発生した場合に該記事の解凍を行う圧縮解凍部を備えてもよい。

この圧縮解凍部は、判定部の判定に沿って適切に選択された記事の圧縮を行う。またアクセス可能性が小さいとして圧縮された記事であっても、アクセスが発生することは当然あり得る。そのような場合、解凍処理を行うことで、アクセスしたユーザに対して記事を適切に提供する。

[0008] 上記した情報処理装置の前記判定部においては、前記圧縮対象ブログに含まれる記事ごとのアクセス可能性の度合いに応じて記事ごとに前記圧縮対象記事とするか否かを判定してもよい。

ブログに属する記事のうちの何れかを圧縮する場合に、何れの記事を圧縮対象とするかを選択することが考えられる。本構成によれば、記事ごとのアクセス可能性に応じた判定を行う。

[0009] 上記した情報処理装置においては、アクセス要求に応じて既に圧縮した記事の解凍が行われた場合、前記判定部は、解凍から所定期間、当該記事を前記圧縮対象記事と判定しなくてもよい。

つまりアクセス時に解凍した記事は、所定期間の間は解凍したままとする。

[0010] 上記した情報処理装置の前記判定部においては、既に圧縮した記事について、同一の前記圧縮対象ブログの他の記事に対するページビューの増加傾向を示す値に応じた解凍の是非を判定し、前記圧縮解凍部は前記解凍の是非に応じて当該記事を解凍しておいてもよい。

或るブログが何らかの人気記事によってアクセス数が顕著に上昇したような場合、そのブログに含まれる別の記事は、これまでアクセスされていなかったとしても、今後アクセスされる可能性が高まる。そこで解凍するように判定する。

[0011] 上記した情報処理装置の前記判定部においては、記事内容に基づいて、既

に圧縮した記事了解凍するか否かを判定してもよい。

例えば圧縮した記事の内容として、或る設定したキーワードや時事語を含む記事、特定のテーマの記事などを抽出し、これらを解凍対象とする。

[0012] 上記した情報処理装置において、前記人気度指数は、ブログ全体についての総ページビュー数、記事ごとのページビュー数、ブログにアクセスしたユニークユーザ数、ブログに設定された総被リンク数、ブログに投稿された総コメント数、ブログにコメントを投稿したユニークユーザ数、ブログのページランクを示す値、ブログに対するアクセス要求が無い期間の長さ、所定期間におけるブログ全体についての総ページビュー数、ブログ全体についてのページビューの増加傾向を示す値、ブログ更新頻度（記事投稿頻度）を示す値、ブログページ上に掲載される広告に対するクリック数、ブログ全体についてのデータ量の増加傾向を示す値、の少なくとも一部に基づいて求められた値とすることが考えられる。

これらの値はブログごとの人気に応じた値となりやすい。

[0013] 本発明に係る情報処理方法は、1または複数の記事が含まれるブログの人気度指数を取得する人気度指数取得ステップと、前記ブログごとの前記人気度指数に応じて該ブログごとに圧縮するか否かを判定するための閾値を設定する閾値設定ステップと、一つの前記ブログに含まれる記事の総データ量と前記閾値とに基づいて当該一つの前記ブログに含まれる記事の少なくとも一部を圧縮する圧縮対象ブログとするか否かを判定し、前記圧縮対象ブログに含まれる各記事の中から圧縮対象とする記事を圧縮対象記事として判定する判定ステップと、を行う。

この情報処理方法により、情報処理装置によって適切に圧縮する記事を判定することができる。

本発明に係るプログラムは、上記各ステップに相当する手順を情報処理装置に実行させるプログラムである。本発明に係る記憶媒体は、上記プログラムを記憶したものである。これらにより上述の情報処理装置の処理を実現する。

発明の効果

[0014] 本発明によれば、圧縮する記事を適切に選択することができるため、閲覧時の解凍処理を可能な限り生じさせないようにしつつ、記憶リソースの有効な利用を図ることができる。

図面の簡単な説明

[0015] [図1]本発明の実施の形態のブログサーバを含むネットワークの説明図である。

[図2]実施の形態で利用できるコンピュータ装置のブロック図である。

[図3]実施の形態のブログサーバの機能構成の説明図である。

[図4]実施の形態の管理データベースの説明図である。

[図5]第1の実施の形態の圧縮判定処理のフローチャートである。

[図6]第1の実施の形態の圧縮処理のフローチャートである。

[図7]実施の形態のアクセス要求受信時の処理のフローチャートである。

[図8]実施の形態の解凍後の処理例のフローチャートである。

[図9]第2の実施の形態の人気度指数算出処理のフローチャートである。

[図10]第2の実施の形態のアクセス可能性指数算出処理のフローチャートである。

[図11]第3の実施の形態の圧縮処理のフローチャートである。

[図12]第4の実施の形態の圧縮判定処理のフローチャートである。

[図13]第4の実施の形態の圧縮解凍判定処理のフローチャートである。

[図14]第5の実施の形態の圧縮解凍処理のフローチャートである。

[図15]第6の実施の形態の解凍判定処理のフローチャートである。

[図16]第6の実施の形態の解凍処理のフローチャートである。

[図17]第7の実施の形態の解凍判定処理のフローチャートである。

[図18]第8の実施の形態の圧縮判定処理のフローチャートである。

発明を実施するための形態

[0016] 以下、実施の形態を次の順序で説明する。

<1. システム構成>

< 2. ブログサーバ及びデータベース >

< 3. 第 1 の実施の形態 >

< 4. 第 2 の実施の形態 >

< 5. 第 3 の実施の形態 >

< 6. 第 4 の実施の形態 >

< 7. 第 5 の実施の形態 >

< 8. 第 6 の実施の形態 >

< 9. 第 7 の実施の形態 >

< 10. 第 8 の実施の形態 >

< 11. まとめ及び変形例 >

< 12. プログラム及び記憶媒体 >

[0017] なお、以下の説明において、“ブログ”とは、ウェブログあるいは単にブログと呼ばれる日記形式のウェブページのことである。より具体的には、ブログサーバはユーザに対してブログを形成する環境（記憶容量やウェブページ）を提供し、ユーザは投稿等の形式で文章や画像による記事を自分のブログにアップロードする。ブログサーバは通常、当該記事を一般（または限定した範囲）の閲覧に供する。但し非公開のものとしてもよい。

記事の内容については特に限定されない。ユーザが情報発信に用いる内容でも、個人的な日記等の内容でもよい。また「ブログ」と呼ばれていないものであっても、同等のものを含む。

[0018] “記事”とは、ブログを構成する要素であって、文章や画像によって構成された一つの単位（例えば投稿単位）のことを指す。内容には関わらない。なお、必ずしも一つの話題としての記事ではなく、1または複数の話題について一つのURLで閲覧される記事群を指すものと考えてもよい。

[0019] “ユーザ”とは、自己のブログに記事を書き込む記述者としてのユーザ（いわゆるブロガー）と、他人または自己のブログを閲覧する閲覧者としてのユーザが想定される。これらを区別して「記述者」「閲覧者」と表記する。もちろん一人のユーザがある時点では記述者となりある時点では閲覧者とな

ることが通常に想定される。

[0020] “圧縮”とは、いわゆるデータ圧縮のことであり、テキストデータや画像データ等の各種データを、そのデータの実質的な性質を保ったまま、データ量を減らした別のデータに変換することである。

“解凍”とは、圧縮されたデータを圧縮前の状態に戻すことである。但し、圧縮時にいわゆる非可逆圧縮が行われた場合など、データが完全に圧縮前の状態に戻らない場合も含む。本明細書では、少なくとも記事の内容の閲覧等が可能な状態とすることを“解凍”ということとする。

[0021] < 1. システム構成 >

図 1 に実施の形態のブログサーバ 1 を含むネットワークシステムの構成例を示す。

本実施の形態に係るネットワークシステムは、ブログサーバ 1 と複数のユーザ端末 5 がネットワーク 2 により相互に通信可能に接続されている。

またブログサーバ 1 は各種データベースにアクセス可能とされている。なお、以下「データベース」については「DB」と表記する。図ではブログサーバ 1 がアクセス可能な DB としてブログ DB 5 1、画像 DB 5 2、管理 DB 5 3 を例示している。

[0022] ネットワーク 2 の構成は多様な例が想定される。例えば、インターネット、イントラネット、エキストラネット、LAN (Local Area Network)、CATV (Community Antenna Television) 通信網、仮想専用網 (Virtual Private Network)、電話回線網、移動体通信網、衛星通信網等が想定される。

またネットワーク 2 の全部または一部を構成する伝送媒体についても多様な例が想定される。例えば IEEE (Institute of Electrical and Electronics Engineers) 1394、USB (Universal Serial Bus)、電力線搬送、電話線等の有線でも、IrDA (Infrared Data Association) のような赤外線、ブルートゥース (登録商標)、802.11 無線、携帯電話網、衛星回線、地上波デジタル網等の無線でも利用可能である。

[0023] ブログサーバ 1 は、ユーザに対するブログサービスの管理運営を行う組織

によって用いられる情報処理装置である。ブログサーバ1は、ユーザ（記述者）へのブログ環境の提供やユーザ（閲覧者）のアクセス要求に応じたブログ記事ページ等のウェブページデータの配信を行う。

具体的にはブログを開設したい記述者に対しては、その記述者のブログとしてのウェブページの設定やユーザ情報の登録などを行う。既にブログを開設済みの記述者に対しては、記述者が投稿する記事の保存を行う。

また一般の閲覧者となるユーザからのアクセス要求に応じて、該当のウェブページに係るウェブページデータを配信する。

このブログサーバ1が、本発明請求項の情報処理装置の実施の形態に相当する。

[0024] ユーザ端末5は、記述者や閲覧者としてのユーザが使用する端末である。このユーザ端末5は、例えば、通信機能を備えたPC（Personal Computer）やフィーチャーフォンやPDA（Personal Digital Assistant）、或いは、スマートフォンやタブレット端末などのスマートデバイスなどが想定される。

ユーザ端末5では、必要に応じて各種の送受信処理や表示処理などが実行される。

閲覧者は、ユーザ端末5においてウェブブラウザを介して、関心のあるブログの閲覧を任意に行うことができる。

記述者は、ユーザ端末5により自分のブログページにアクセスし、閲覧したり、新規に記事を投稿したりすることができる。

ユーザ端末5ではこれらの動作のための通信処理や表示処理等を行うことになる。

[0025] 図1に示したブログサーバ1やユーザ端末5を構成する情報処理装置のハードウェア構成を図2に示す。ブログサーバ1やユーザ端末5として示した各装置は、情報処理および情報通信が可能な図2に示すようなコンピュータ装置によって実現される。

[0026] 図2において、コンピュータ装置のCPU（Central Processing Unit）1

１０１は、ＲＯＭ（Read Only Memory）１０２に記憶されているプログラム、または記憶部１０８からＲＡＭ（Random Access Memory）１０３にロードされたプログラムに従って各種の処理を実行する。ＲＡＭ１０３にはまた、ＣＰＵ１０１が各種の処理を実行する上において必要なデータなども適宜記憶される。

ＣＰＵ１０１、ＲＯＭ１０２、およびＲＡＭ１０３は、バス１０４を介して相互に接続されている。このバス１０４には、入出力インタフェース１０５も接続されている。

入出力インタフェース１０５には、入力装置１０６、出力装置１０７、記憶部１０８、通信部１０９が接続されている。

入力装置１０６はキーボード、マウス、タッチパネルなどにより構成される。

出力装置１０７はＬＣＤ（Liquid Crystal Display）、ＣＲＴ（Cathode Ray Tube）、有機ＥＬ（Electroluminescence）パネルなどよりなるディスプレイ、並びにスピーカなどにより構成される。

記憶部１０８はＨＤＤ（Hard Disk Drive）やフラッシュメモリ装置などにより構成される。

通信部１０９はネットワーク２を介しての通信処理や機器間通信を行う。

入出力インタフェース１０５にはまた、必要に応じてメディアドライブ１１０が接続され、磁気ディスク、光ディスク、光磁気ディスク、或いは半導体メモリなどのリムーバブルメディア１１１が適宜装着され、リムーバブルメディア１１１に対する情報の書込や読出が行われる。

[0027] このようなコンピュータ装置では、通信部１０９による通信によりデータやプログラムのアップロード、ダウンロードが行われる。またリムーバブルメディア１１１を介したデータやプログラムの受け渡しが可能である。

ＣＰＵ１０１が各種のプログラムに基づいて処理動作を行うことで、ブログサーバ１やユーザ端末５としての必要な情報処理や通信が実行される。

なお、ブログサーバ１やユーザ端末５を構成する情報処理装置は、図２の

ようなコンピュータ装置が単一で構成されることに限らず、複数のコンピュータ装置がシステム化されて構成されてもよい。複数のコンピュータ装置は、LAN等によりシステム化されていてもよいし、インターネット等を利用したVPN等により遠隔地に配置されたものでもよい。複数の情報処理装置には、クラウドコンピューティングサービスによって利用可能なサーバ群（クラウド）としての情報処理装置が含まれてもよい。

[0028] <2. ブログサーバ及びデータベース>

図3に1または複数の情報処理装置で構成されるブログサーバ1としての機能構成および各種のDBを示す。

ブログサーバ1としての各機能は、情報処理装置においてCPU101でプログラムに応じて実行される処理により実現される機能である。但し以下説明する全部または一部の各構成の処理をハードウェアにより実現してもよい。

また各機能をソフトウェアで実現する場合に、各機能がそれぞれ独立したプログラムで実現される必要はない。一つのプログラムにより複数の機能の処理が実行されてもよいし、一つの機能が複数のプログラムモジュールの連携で実現されてもよい。

また各機能は複数の情報処理装置に分散されていてもよい。更に機能の一つが、複数の情報処理装置によって実現されてもよい。

[0029] ブログサーバ1は、図示するようにブログ管理部11、人気度指数取得部12、閾値設定部13、判定部14、圧縮解凍部15としての機能を備える。

[0030] ブログ管理部11は、ブログサービスを提供するサーバとして必要な処理を実行する。例えばユーザへのブログ環境の提供、記述者としてのユーザの情報の管理、作成されたブログの記憶管理、各ブログに関する情報管理、アクセス要求に応じたブログ（記事）のウェブページの配信などを行う。

またブログ管理部11は管理DB53の情報の更新や読み出しを逐次行う。

[0031] 人気度指数取得部 12 は、ブログ内の記事について圧縮するか否かの判定を行う際などに、その判定に用いる情報として、ブログの人気度を示す人気度指数を取得する処理を行う。人気度指数の取得処理では、人気度指数を示す値そのものを取得してもよいし、複数の指標から人気度指数を算出することによって取得してもよい。

[0032] ブログについての人気度指数は、ブログ全体についての総ページビュー数、記事ごとのページビュー数、ブログにアクセスしたユニークユーザ数、ブログに設定された総被リンク数、ブログに投稿された総コメント数、ブログにコメントを投稿したユニークユーザ数、ブログのページランクを示す値、ブログに対するアクセス要求が無い期間の長さ、所定期間におけるブログ全体についての総ページビュー数、ブログ全体についてのページビューの増加傾向を示す値、ブログ更新頻度を示す値、ブログページ上に掲載される広告に対するクリック数、ブログ全体についてのデータ量の増加傾向を示す値などのうちから、1 または複数の値に基づいて算出される。

[0033] 例えば、ブログ全体についての総ページビュー数に基づいて人気度指数を算出してもよいし、ブログ全体についての総ページビュー数に加えてブログに投稿された総コメント数を加味して算出してもよい。

[0034] 閾値設定部 13 は、ブログに含まれる少なくとも一部の記事を圧縮する圧縮対象ブログとするか否かを判定（詳しくは後述）するための閾値をブログごとに設定する処理を行う。

閾値の設定には人気度指数取得部 12 が取得した人気度指数を用いる。人気度指数が高い（人気が高い）ブログほど閾値を高く設定することで、圧縮処理が生じにくくすると共に解凍処理も生じにくくする。

なお、以降の記述においては、上記閾値を「圧縮対象ブログ判定閾値」と記載する。

[0035] 判定部 14 は、圧縮対象ブログ判定閾値に基づきブログに含まれる記事を圧縮するか否かを判定する処理を行う。換言すれば、当該ブログを圧縮対象ブログとするか否かを判定する処理を行う。判定部 14 は、圧縮対象ブログ

と判定した場合には、そのブログに含まれる記事ごとに圧縮するか否かの判定も行う。

また判定部 14 は、圧縮された記事について、人気度指数に基づいて解凍するか否かを判定する処理や、内容に応じて解凍するか否かを判定する処理や、ブログの人気傾向などに応じて解凍するか否かを判定する処理なども行う。

具体例は各実施の形態の処理として後述する。

[0036] 圧縮解凍部 15 は、判定部 14 によって圧縮すると判定された記事を圧縮する処理を行う。

また圧縮解凍部 15 は、既に圧縮された記事に対してのアクセス要求が発生した場合に、圧縮に対する解凍処理を行う。

更に圧縮解凍部 15 は、判定部 14 によって解凍すると判定された圧縮済みの記事を解凍する処理を行う。

なお、記事の圧縮処理を行った結果、それ以上の記事の圧縮処理が不要となったブログは、圧縮対象ブログではなくなる。

[0037] 図 3 にはブログサーバ 1 がアクセスする DB としてブログ DB 51、画像 DB 52、管理 DB 53 を示している。

ブログ DB 51 は、各記述者についてのブログデータをウェブページデータとして保存する DB である。各ブログについては、記述者の投稿に応じて記事が追加されていく。

ブログを形成するウェブページのデータは、例えば、HTML (HyperText Markup Language) や XHTML (Extensible HyperText Markup Language) などの構造化文書ファイルである。構造化文書ファイルには、記述者が投稿した記事のテキストデータや各種画像等の画像データの指定情報と、それらの配置や表示態様（文字色やフォントや大きさや装飾など）が記述されている。

またブログに対しては閲覧者がコメントを投稿することもできる。そのような閲覧者からのコメントデータもブログやブログ内の個々の記事に紐づけ

てブログＤＢ５１に保存される。

ブログサーバ１は、ユーザ端末５から或るブログについてのアクセス要求があった場合に、要求されたブログページをブログＤＢ５１から読み出してユーザ端末５に配信することになる。

[0038] 画像ＤＢ５２は、ブログに添付された画像データ（静止画データや動画データ）を保存するＤＢである。

ブログ内の記事には、画像を添付することができるが、例えばブログＤＢ５１には記事データ及び記事データに対応した画像の指定情報（リンク情報）が記憶される。そして画像データ自体は画像ＤＢ５２に保存される。

画像が添付されたブログ記事へのアクセス要求の場合、そのウェブページデータがユーザ端末５においてブラウザにより表示されるが、その際、ユーザ端末５はウェブページ上のリンク設定により画像データをブログサーバ１に要求する。ブログサーバ１は当該要求に応じて画像データを画像ＤＢ５２から読み出し、ユーザ端末５に配信する。これによりユーザ端末５上で、画像付きのブログ記事が表示される。

なおこれは一例であり、予め画像データを含むウェブページデータをブログＤＢ５１に格納するようにしてもよい。

[0039] 管理ＤＢ５３は、各ブログを管理するための情報を格納するＤＢである。管理ＤＢ５３の内容の一例を図４に示す。

一つのブログについてはブログＩＤ（Identification）が設定され、ブログＩＤにより付随する情報が管理される。例えばブログ（ブログＩＤ）ごとに、ユーザ情報、ブログ管理情報、ブログ実績情報、サイズ情報、判定情報、圧縮解凍情報、圧縮記事タグなどが、逐次更新されながら管理される。

[0040] ユーザ情報は、ブログを開設した記述者としてのユーザ（ブログ管理者）の情報である。例えばユーザ情報としては、ユーザＩＤ、管理者としてのログインパスワード、ユーザの住所、氏名、年齢等の属性情報、管理者としてのログイン日時、などの情報が含まれる。

[0041] ブログ管理情報は、ブログ自体の属性情報である。例えばブログのＵＲＬ

(Uniform Resource Locator)、ブログのジャンル情報、ブログの開設日時、ブログに含まれる記事数、更新日時情報、ブログのレイアウト情報、リンク設定情報等が含まれる。

[0042] ブログ実績情報としては、ブログの人気の指標となる情報（人気度指数）や、各記事のアクセス可能性の指標となる情報が記憶される。

具体的には、ブログ全体について、総ページビュー数、アクセスしたユニークユーザ数、総被リンク数、ブログのページランクを示す値、ブログに対するアクセス要求が無い期間の長さ、所定期間におけるブログ全体についての総ページビュー数、ブログ全体についてのページビューの増加傾向を示す値、ブログ更新頻度を示す値、ブログページ上に掲載される広告に対するクリック数、ブログ全体についてのデータ量の増加傾向を示す値などが記憶され、逐次更新される。

これらの値はブログごとの人気度に応じた値となるため、人気度指数算出の際に用いる値として適している。なお、算出された人気度指数もブログ実績情報として記憶される。

[0043] また、ブログ実績情報として、記事ごとに、ページビュー数、アクセスしたユニークユーザ数、総被リンク数、記事のランクを示す値、記事に対するアクセス要求が無い期間の長さ、所定期間における記事についてのページビュー数、記事についてのページビューの増加傾向を示す値、記事ページ上に掲載される広告に対するクリック数などが記憶され、逐次更新される。

これらの値は、記事ごとのアクセス可能性に応じた値となるため、アクセス可能性の度合い（以降、アクセス可能性指数）を算出する際に用いる値として適している。なお、算出されたアクセス可能性指数もブログ実績情報として記憶される。

[0044] サイズ情報は、そのブログ全体に含まれる記事の総データ量の情報である。また各記事のサイズ情報を記憶してもよい。サイズ情報は、ブログの更新に応じて更新される。

なお、サイズ情報としてブログDB51に記憶したデータ量と画像DB5

2に記憶した画像データのデータ量を合わせて管理してもよいし、それぞれ別個に管理してもよい。

[0045] 判定情報は、そのブログを圧縮対象ブログとするか否かの判定情報や、各記事について判定部14が人気度指数やその他の情報に基づいて圧縮するか否かを判定した情報である。また圧縮した記事について解凍するか否かを判定部14が判定した情報も含む。即ち圧縮可否や解凍可否を示す情報である。これらは例えばフラグデータとして更新される。

圧縮可否情報として圧縮可を示す情報が記憶された記事は、圧縮処理の対象とされた記事であることを示す。そして、圧縮処理の後には当該フラグデータがクリアされ、圧縮否（即ち圧縮処理の対象とはしないことを示す状態）を示す情報が上書きされる。

他にも、判定情報としてブログごとの圧縮対象ブログ判定閾値が記憶される。

[0046] 圧縮解凍情報は、ブログ内の各記事について、オリジナル状態／圧縮状態／圧縮から解凍した状態など、圧縮や解凍の実行状況を示す情報である。圧縮解凍情報は、これらを識別するステータス情報として構成されればよい。

また圧縮や解凍の履歴情報として、圧縮や解凍の実行日時も記憶される。

以降の記載においては、オリジナル状態の記事を「未圧縮記事」、圧縮状態の記事を「圧縮記事」、圧縮から解凍した状態の記事を「解凍記事」と記載する。

なお、解凍記事のデータは、未圧縮記事のデータと同一である場合もあるが、非可逆圧縮を用いているために元の記事データよりも低品質となる場合もある。つまり必ずしも未圧縮記事と解凍記事のデータは同一ではない。そこで圧縮後に解凍した記事データについては「未圧縮記事」と区別するために「解凍記事」と表記する。

[0047] 圧縮記事タグは、各記事について内容に応じて設定されたタグである。

例えば記事に出現するキーワード的な語句、時事的な語句、記事のジャンルなどがタグとして設定され、登録される。例えば圧縮を行う際に記事内容

に応じたタグが作成されて圧縮記事タグとして登録される。圧縮記事タグは、解凍せずに圧縮記事の内容を推定するために利用される。

[0048] 以上の各DB（ブログDB 51、画像DB 52、管理DB 53）は、ブログサーバ1がアクセス可能とされていればどのような形態で実現されていてもよい。例えばブログサーバ1と同一システム内の記憶部に各DBのすべてが形成されていてもよいし、各DBの一部または全部が別体、遠隔地等のコンピュータシステムに設けられていてもよい。もちろん各DBが一つの装置（例えば一つのHDD等）内に形成されている必要はない。また各DBのそれぞれが、それぞれ一つのDBとして構成される必要もない。例えば管理DB 53として記憶される情報が、複数のDB（例えばブログに関するユーザ管理用のDBとブログ管理用のDBなど）により記憶管理されてもよい。以上の各DBは、実施の形態の処理に関連する情報の記憶部を、それぞれ一つのDBの形態で例示したものに過ぎない。

[0049] <3. 第1の実施の形態>

ブログサーバ1が実行する第1の実施の形態としての処理例を説明していく。

現在、一般ユーザにとってブログは容易に始められるが、少しの記事をアップロードした後、飽きてしまうユーザもいれば、長く続ける人もいる。また、アクセス要求の多い人気ブログもあれば、ほとんど閲覧者がいないブログもある。

ブログサーバ1としては、これら多様なユーザに対して、分け隔て無くブログを維持する必要があるが、そのために記憶リソースの負担が大きくなりがちである。

そこで本実施の形態において、ブログサーバ1は、ブログごとの人気度指数に基づいて、圧縮するブログ及び記事を決定する。また、圧縮記事に対するアクセス要求があった場合には、解凍して配信する。

[0050] 但し、圧縮処理・解凍処理もある程度の処理負担がかかるため、あまり頻繁に行いたくない。また、圧縮した記事を解凍して配信することは、処理負

担と共に応答時間も増えることになり、ユーザにパフォーマンス低下を感じさせる可能性もあるため、なるべく圧縮した記事へのアクセス要求が発生しないようにしたい。

そこで本実施の形態では、圧縮対象ブログに属する各記事の中でも、アクセス可能性が小さい記事をよりの確に選択して圧縮対象記事とする。

[0051] 図5はブログサーバ1が実行する圧縮判定処理の例を示している。圧縮判定処理では、圧縮対象ブログを判定すると共に、当該圧縮対象ブログに含まれる記事の中から圧縮対象記事を判定する。

なお、この例では、ブログ実績情報の中の一つの値（指標）をそのまま人気度指数として用いる。また、記事ごとのアクセス可能性指数についても、ブログ実績情報の中の一つの値をそのままアクセス可能性指数として用いる。

図5以降、後述する図18までのフローチャートで示す各処理は、ブログサーバ1が図3に示したブログ管理部11、人気度指数取得部12、閾値設定部13、判定部14、圧縮解凍部15としての機能により実行する処理である。

[0052] ブログサーバ1は図5の圧縮判定処理を、逐次、ブログDB51に保存している全部または一部のブログに対して実行する。これは、今回圧縮対象とするブログを圧縮対象ブログとして判定すると共に、当該圧縮対象ブログに含まれる記事のうち何れの記事を圧縮対象記事とするかを判定する処理である。

まず、ステップS101でブログサーバ1は、圧縮判定処理の対象とする一つのブログを特定する。例えばブログID順に従って一つのブログを選択するものとすればよい。

[0053] ステップS102でブログサーバ1は、処理対象として特定したブログについての人気度指数を取得する。本例では、管理DB53にブログ実績情報として記憶された情報から人気度指数として採用する情報を一つ取得する処理である。具体的には、ページビュー数、アクセスしたユニークユーザ数、

総被リンク数、ブログのページランクを示す値、ブログに対するアクセス要求が無い期間の長さ、所定期間におけるブログ全体についての総ページビュー数、ブログ全体についてのページビューの増加傾向を示す値、ブログ更新頻度を示す値、ブログページ上に掲載される広告に対するクリック数、ブログ全体についてのデータ量の増加傾向を示す値などの中から、人気度指数として採用する情報を一つ選択して取得する。

なお、人気度指数は他のブログに対する相対的な人気度を示すものであるため、ブログごとに異なる指標を取得することは好ましくない。即ち、人気度指数としてブログの総ページビュー数を採用する場合は、何れのブログに対する人気度指数を取得する場合であってもそれぞれのブログの総ページビュー数を取得する。

[0054] 次に、ステップS 103でブログサーバ1は、人気度指数に応じた圧縮対象ブログ判定閾値を決定する処理を実行する。

例えば、一人当たりの記述者に割り当てられているブログの総データ量が2GB（Giga Byte）である場合に、圧縮対象ブログ判定閾値は、2GBよりも少ない値とされる。一例として、圧縮対象ブログ判定閾値を人気度指数の高／中／低に基づいて1.7GB／1.5GB／1.2GBのように決定する。

人気度指数を高／中／低に分類する一例を説明する。

例えば、人気度指数としてブログの総ページビュー数を採用し、

- ・ 総ページビュー数が0～10000＝人気度指数が低い
- ・ 総ページビュー数が10,001～1,000,000＝人気度指数が中くらい
- ・ 総ページビュー数が1,000,001以上＝人気度指数が高い

とする。このとき、総ページビュー数が50,000のブログは、人気度指数が中であるとして、圧縮対象ブログ判定閾値が1.5GBに設定される。

なお、人気度指数によって圧縮対象ブログ判定閾値を三段階に分けるのはあくまで一例であり、2段階に分けてもよく、4段階以上に分けてもよい。

[0055] 圧縮対象ブログ判定閾値を4段階以上に分類する一例として、数式を用いる例を説明する。

例えば、全てのブログの中で最大の総ページビュー数が10万（＝人気度指数の最大が10万）のブログであり、今回の処理対象とされたブログの総ページビュー数が3万（人気度指数が3万）である場合に、圧縮対象ブログ判定閾値＝ $1.2\text{GB} + (3\text{万} / 10\text{万} \times 0.5\text{GB}) = 1.35\text{GB}$ などのように算出してもよい。この数式によれば、人気度指数＝10万のブログの圧縮対象ブログ判定閾値は1.7GBとなり、人気度指数＝0万のブログの圧縮対象ブログ判定閾値は1.2GBとなる。即ち、圧縮対象ブログ判定閾値は、人気度指数に応じて細かく設定される。なお、この場合には、人気度指数を高／中／低の3段階などに分ける必要は無い。

他にも、圧縮対象ブログ判定閾値は、全てのブログの人気度指数の分布を加味して算出してもよいし、処理対象とされたブログの人気度指数のみを用いて算出してもよい。

[0056] 続いて、ステップS104でブログサーバ1は、処理対象のブログにおける総データ量が圧縮対象ブログ判定閾値を超えているか否かを判定する処理を実行する。

ブログにおける総データ量が圧縮対象ブログ判定閾値を超えていない場合、ブログサーバ1は後述するステップS111の処理へと遷移する。

一方、ブログにおける総データ量が圧縮対象ブログ判定閾値を超えている場合、ブログサーバ1は、処理対象のブログを圧縮対象ブログと判定し、当該判定結果を管理DB53の判定情報として記憶した後、ブログ内の各記事について圧縮可否を判定するためにステップS105～S110の各処理を行う。

[0057] ステップS105でブログサーバ1は、ブログ内の一つの記事を選択する。

ステップS106でブログサーバ1は、選択した記事が既に圧縮済であるか否かを確認する。これは例えば管理DB53における圧縮解凍情報を参照

すればよい。

もし圧縮済であれば、当該記事の圧縮可否判定は不要なためステップS 1 1 1に進む。ステップS 1 1 0で全ての記事について処理を終えたか否かを確認し、終えていなければステップS 1 0 5に戻って次の記事を選択する。

なお、ステップS 1 1 1の全記事とは、今回処理対象とする記事の全てという意味である。ブログ内の全ての記事という場合もあるし、ブログ内の一部（例えば特定の期間に投稿された記事など）でもよい。

[0058] ステップS 1 0 5で処理対象に選択した記事が圧縮されていない状態の記事である場合、ブログサーバ1はステップS 1 0 6からS 1 0 7に進み、当該記事についてのアクセス可能性指数としてのブログ実績情報を取得する。本例におけるアクセス可能性指数は、管理DB 5 3にブログ実績情報として記憶された情報からアクセス可能性指数として採用する情報を一つ取得する処理である。具体的には、管理DB 5 3に記憶された当該記事についてのページビュー数、アクセスしたユニークユーザ数、総被リンク数、記事のランクを示す値、記事に対するアクセス要求が無い期間の長さ、所定期間における記事についてのページビュー数、記事についてのページビューの増加傾向を示す値、記事ページ上に掲載される広告に対するクリック数などの中から、アクセス可能性指数として採用する情報を一つ選択して取得する。

なお、アクセス可能性指数に関しても人気度指数と同様に、記事ごとに同じ指標を取得することが望ましい。

[0059] 続いて、ブログサーバ1はステップS 1 0 8で、当該記事についての圧縮可否判定をアクセス可能性指数に基づいて行う。即ち、アクセス可能性が大きければ圧縮すべきでないと判定し、アクセス可能性が小さければ圧縮してもよい記事（或いは圧縮すべき記事）と判定する。

例えば、アクセス可能性指数として当該記事に対するアクセス要求が無い期間の長さを選択した場合、当該期間長が3年以上であるか未満であるかを判定し、3年以上アクセス要求が無い記事はアクセス可能性が低いとして圧縮対象記事と判定する。なお、3年という数値（判定のための閾値）はあく

まで一例である。この数値は記述者ごとに変えてもよいし、一律同じ値であってもよい。記述者ごとに変える場合は、処理対象のブログに属する各記事のアクセス可能性指数の分布を鑑みて決定することが望ましい。即ち、アクセス可能性指数が低い記事が多いブログであれば、上記「3年」という判定のための閾値を「5年」のように長くすることが考えられるし、アクセス可能性指数が高い記事が多いブログであれば、上記「3年」という判定のための閾値を「1年」のように短くすることが考えられる。

[0060] このように、まずステップS103及びS104においてブログの人気度指数を考慮して圧縮対象ブログを判定した後、ステップS107及びS108においてアクセス可能性指数を考慮して圧縮可否を判定する。

これによってブログ自体の人気を考慮したうえで、あまりアクセスされていない記事の圧縮可否を判定できることになる。

例えば単純に、「記事に対して1度もアクセスがない期間が3年以上続いていること」のように、記事に対して1度もアクセスがない期間を指標値として、指標値が一定の条件を満たす記事について圧縮することも考えられるが、この場合、必ずしもアクセス可能性が適切に判断されているとはいえない面がある。

例えば、全体として人気のあるブログの複数の記事のうちで、3年の間に1度もアクセスがない記事と、全体として不人気なブログの複数の記事のうちで、3年の間に1度もアクセスがない記事とでは、次にアクセスが発生する可能性は前者の方が高いと考えられる。すなわち、単に記事のみのアクセス可能性の指標のみでは、実際のアクセス可能性をよりの確に判定することが難しい。換言すれば、人気ブログも不人気ブログも同様の条件で圧縮する記事を判定することは、なるべく圧縮や解凍の機会を少なくしたいという観点からは、必ずしも妥当ではない。

そこでステップS104でブログサーバ1は、ブログの人気を反映させた圧縮対象ブログ判定閾値を用いることにより、人気ブログが圧縮対象ブログとなり難いようにしている。

[0061] 続くステップS 1 0 9ではブログサーバ1は、当該記事についての圧縮可否判定の結果を判定情報として記憶する。例えば、当該記事についての判定情報として管理DB 5 3に記憶された圧縮可／否のフラグを更新または維持する。

[0062] 以上で一つの記事について圧縮可否判定を終えたら、ブログサーバ1はステップS 1 1 0で、現在の処理対象のブログについて今回処理対象としている記事の全てについて判定を終えたか否かを確認し、終えていなければステップS 1 0 5に戻って次の記事を選択する。そしてステップS 1 0 6～S 1 0 9を実行する。

[0063] 或るブログについて今回対象の全ての記事について圧縮可否判定を終えたら、ステップS 1 1 1で、続いて他のブログについても処理を実行するかを確認する。他のブログについても同様の処理を行う場合は、ステップS 1 0 1に戻って、他の一つのブログを処理対象として特定し、以下同様の処理を行う。

例えば今回処理対象とする全てのブログについて処理を終えていた場合は、ステップS 1 1 1から図5の圧縮判定処理を終える。

[0064] ブログサーバ1は、この図5のような圧縮判定処理を逐次実行する。例えば定期的に、ブログDB 5 1に保存されている全てのブログについて実行することが考えられる。これにより、各ブログについて、各記事が、圧縮してよいものか否かが判定され、その判定情報が管理DB 5 3に記憶される。

なお、各ブログについて、図5のような圧縮判定処理をある時点で1回行うだけで無く、期間をおいてくり返し実行することが望ましい。時期ごとに記事のアクセス状況は変動し、また所定期間以上アクセスがなかった記事も発生することが想定されるからである。

[0065] ブログサーバ1は以上のような圧縮判定処理を適宜行うとともに、図6に例示する圧縮処理を適宜行う。例えば全部または一部のブログを対象として定期的に圧縮処理を行う。

図6の圧縮処理としてブログサーバ1は、まずステップS 2 0 1で処理対

象のブログを一つ特定する。

ステップS 2 0 2でブログサーバ1は、処理対象と特定したブログについての判定情報を取得する。即ち管理DB 5 3において当該ブログのブログIDに対応して記憶されている判定情報である。具体的には例えば、図5の圧縮判定処理において判定した圧縮対象ブログであるか否かを示す情報や、記事ごとの圧縮可否のフラグ情報を確認する処理となる。

判定情報により、当該ブログが圧縮対象ブログであるか否か、及び、ブログにおける各記事について圧縮可とされているか否かを確認できる。

そこでステップS 2 0 3でブログサーバ1は、処理対象のブログが圧縮対象ブログであるか否かを判定する。

[0066] もし処理対象のブログが圧縮対象ブログでなければ、ブログサーバ1はステップS 2 0 3からS 2 1 0に進み、当該ブログについての圧縮処理を終了する。そして続いて他のブログについても圧縮処理を実行するかを確認する。他のブログについても圧縮処理を行う場合は、ステップS 2 0 1に戻って、他の一つのブログを処理対象として特定する。

[0067] 圧縮対象とされた1以上の記事が存在する場合は、ブログサーバ1はステップS 2 0 3から2 0 4に進み、圧縮対象ブログに属する各記事から、圧縮対象記事を特定する。複数の記事が圧縮対象記事とされている場合は、全ての圧縮対象記事を特定する。

[0068] 続いて、ブログサーバ1はステップS 2 0 5において、記事圧縮を行う。即ち、ステップS 2 0 4で特定した1または複数の記事のデータ圧縮を行う。そして圧縮データ化された記事を、当該ブログに対応づけてブログDB 5 1や画像DB 5 2に記憶する。

[0069] このステップS 2 0 5でどのような圧縮を行うかは多様に考えられる。

まず圧縮対象部分の設定、即ち記事データにおけるどの部分を圧縮するかという種別として、

- ・記事におけるテキストデータと画像データの両方を圧縮する
- ・記事におけるテキストデータの全部を圧縮する

- ・記事におけるテキストデータの一部を圧縮する
- ・記事に含まれる画像データの全部を圧縮する
- ・記事に含まれる画像データの一部を圧縮する

ということが想定される

[0070] 記事のテキストデータと画像データの両方を圧縮することによれば、圧縮効果を高くすることができ、必要記憶容量の削減効果を高くすることができる。

記事におけるテキストデータの全部を圧縮することによれば、テキストデータ量や圧縮率にもよるが、圧縮効果（容量削減効果）を高くすることができる。特に記事内容としてテキストデータが中心であるブログで有効である。

テキストデータの一部を圧縮することによれば、もし圧縮後にアクセスが生じたときの配信対応を迅速化することも可能である。例えばブログの後半（閲覧時のファーストビューに現れない部分）を圧縮しておく。圧縮部分は後述のように解凍して配信することが想定されるが、ファーストビュー部分は圧縮しないことで、ユーザ端末5に対して迅速に（解凍処理を経ずに）配信できる。そしてユーザ端末5でファーストビュー表示を行っている間に後続部分の解凍を行って配信すれば、ユーザにとって応答の遅れが生じていないように感じさせることができる。

またテキストデータのみの圧縮を行うことは、画像データ圧縮を行う場合に比較して処理負担が小さく、処理時間も短いという利点も得られる。

[0071] 記事における画像データの全部を圧縮することによれば、データ量の大きい部分であるため、圧縮効果（容量削減効果）を高くすることができる。画像データの圧縮は、画像の解像度を低下させる圧縮とすれば、容量削減効果は特に高い。複数の画像データが存在する場合、全部の画像データに限らず、一部の画像データを圧縮するものでもよい。

記事における一部の画像データを圧縮する場合、閲覧時のファーストビューに現れない画像を選んで圧縮するとよい。その場合、もし圧縮後にアクセ

スが生じたときには、まず解凍不要な画像データを配信すればよいため、ユーザにとって応答の遅れが生じていないように感じさせることができる。そしてユーザ端末5でファーストビュー表示を行っている間に後続の画像データを解凍して配信すればよい。

[0072] 以上のような記事内における圧縮対象部分の設定は、固定的でもよいし、状況に応じて変更できるようにしてもよい。例えばブログDB51や画像DB52の記憶リソース状況などに応じて自動的に選択できるようにしてもよい。

例えばブログDB51の記録可能なリソースが所定量以下になったら、テキストデータ圧縮を選択し、画像DB52の記録可能なリソースが所定量以下になったら画像データ圧縮を選択する。ブログDB51と画像DB52の両方の記憶可能容量が低下した状態であれば、テキストデータと画像データの両方を圧縮するなどである。

[0073] また、圧縮対象部分の設定は、ブログごとや記事ごとに自動的に選択するようにしてもよい。

記事内容に応じて圧縮処理内容を決定する例としては、

- ・記事においてテキストデータが所定量以上ならテキストデータのみの圧縮を行い、所定量未満ならテキストデータと画像データの全体圧縮を行う。
 - ・記事において画像データが存在すれば画像データのみの圧縮を行う。
- などである。

更にブログごとに圧縮対象部分を選択する例としては、ブログの全体のテキスト／画像の比率からテキスト中心ブログか画像中心ブログかを判定し、テキスト中心ブログの場合はテキストデータの圧縮、画像中心ブログの場合は画像データの圧縮を行うということも考えられる。

逆に、アクセス時にユーザが感じる配信速度を重視する場合、テキスト中心ブログの場合は画像データの圧縮、画像中心ブログの場合はテキストデータの圧縮を行うということも考えられる。

[0074] なお、画像データとして動画が含まれている場合、更に動画圧縮と音声圧

縮の両方、一方を選択することも考えられる。

[0075] 以上のような圧縮対象部分の設定のほかに、圧縮方式の設定も多様に考えられる。公知の通り、画像データやテキストデータの圧縮には多様な方式が存在し、また圧縮率も多様に選択できる。可逆圧縮、非可逆圧縮という選択も可能である。

この圧縮方式についても、或る圧縮方式を固定的に用いてもよいし、状況に応じて選択するようにしてもよい。

例えばブログDB51や画像DB52の記録可能なリソースが所定量以下になったら、より圧縮率の高い圧縮方式に切り替えるということが考えられる。

またブログごとや記事ごとに自動的に圧縮方式を選択するようにしてもよい。

例えば人気が低いブログほど圧縮率が高くなるようにしたり、記事のアクセス可能性の低さの程度によって圧縮率の異なる圧縮方式を選択するなどである。

[0076] ブログサーバ1は図6のステップS205で、現在処理対象のブログにおける圧縮対象の1または複数の記事について圧縮処理を行った後、圧縮処理の対象となった記事に関する圧縮可否の情報（フラグ情報）を更新（圧縮可→圧縮否へ更新）する。

続いて、ブログサーバ1はステップS206で管理DB53における圧縮解凍情報を更新する。ここでは当該ブログ内の圧縮を行った記事について、圧縮状態であることを示すように例えばフラグ情報を更新する。また圧縮履歴を追加する。

[0077] ステップS207でブログサーバ1は、圧縮した各記事についてタグ設定を行う。ここでいうタグとは、記事内容を示すキーワードや記事のジャンル情報などを示す情報で、記事の検索、抽出に利用できるようにするものである。

圧縮記事については、記事を対象とするテキスト検索がやりにくくなる。

即ち圧縮記事も検索範囲に含めるようにしたい場合、検索時に、わざわざ解凍を行わなければならない。そこで、タグを設定して登録しておく。

ステップS207の処理を実行時は、処理対象の記事について、圧縮記事と圧縮前の元の記事（即ち未圧縮記事）との双方が記憶されている状態である。

従って、ステップS207でブログサーバ1は、未圧縮記事のデータから、頻出語の抽出、品詞解析による名詞抽出、ジャンル情報の取得などを行い、タグとして登録する1または複数の語句を設定する。

そしてブログサーバ1ステップS208で圧縮記事タグとして管理DB53に登録する。即ち圧縮した記事のそれぞれに対応させて、キーワード等の1または複数の語句を登録する。

[0078] なお、この圧縮記事タグの設定及び登録は、図6の例の場合、或る記事を圧縮した際に、その記事について行うようにしているが、予め全ての記事について行っておいてもよい。特に記事の検索のために、タグを用いており、前記事についてタグ登録を行っているのであれば、タグを本実施の形態の圧縮記事タグとして用いればよい。このステップS207、S208を実行する必要は無い。

但し、特に全記事についてタグ登録を行っていないシステムの場合、図6のステップS207、S208として圧縮記事タグ登録を行うことで、必要な記事についての必要最小限の処理となるため、ブログサーバ1の処理負担の増大防止に有効である。

[0079] ステップS209でブログサーバ1は、圧縮した記事について、元の圧縮前の記事データを削除する。もちろん記事の一部のみの圧縮を行った場合は、圧縮した部分のみの元のデータを削除する。また、解凍記事を再圧縮する処理を行った場合は、元の圧縮前の記事データとしての解凍記事を削除する。

[0080] 以上の処理により一つのブログについての圧縮処理を終えたら、ブログサーバ1はステップS210で他に処理対象となっているブログがあるか否か

を確認する。

そして今回処理対象とする全てのブログについて処理を終えていた場合は、ステップS 2 1 0から図6の圧縮処理を終える。

以上の図6の圧縮処理を行うことで、図5の圧縮判定処理で圧縮可と判定された記事についての実際の圧縮処理が行われ、記憶リソースの回復が図られる。

[0081] 続いて、ブログやブログ内の記事に対するアクセス要求があった場合のブログサーバ1の処理を図7，図8で説明する。

ステップS 3 0 1において、ユーザ端末5からのアクセス要求の受信を確認したブログサーバ1は、続くステップS 3 0 2において、まず要求された記事が、圧縮記事であるか否かを判定する。

圧縮記事でなければ、ブログサーバ1はステップS 3 0 2からS 3 0 3に進み、通常に要求された記事を配信する。即ち該当の記事のウェブページデータをブログDB 5 1から読み出し、ユーザ端末5に送信する。これによりユーザ端末5を使用している閲覧者は所望の記事を閲覧することができる。

[0082] アクセス要求された記事が圧縮記事であった場合、ブログサーバ1はステップS 3 0 4に進み、解凍処理を行う。即ち該当の記事の圧縮状態のデータをブログDB 5 1から読み出し、解凍処理を行う。そしてステップS 3 0 5で解凍後のウェブページデータをユーザ端末5に送信する。これにより、圧縮されていた記事であっても、ユーザ端末5を使用している閲覧者は所望の記事を閲覧することができる。

[0083] なお、上述のように記事データの一部、特にウェブページデータにおけるファーストビューとして現れる領域以外のデータを圧縮するようにしていた場合、ブログサーバ1は、まず非圧縮記事部分をユーザ端末に送信しておき、その間に圧縮部分の解凍処理を行って、解凍でき次第、送信を行うという手法をとることができる。このようにすると解凍処理の時間を閲覧者に感じさせないような配信を実行でき、ブログサーバ1のサービスとしてのパフォーマンス維持が実現できる。

またファーストビュー以外の部分の圧縮に限らず、記事の一部を圧縮している場合は、記事内の非圧縮部分を先に送信するようにすることが同様の理由で望ましい。

[0084] 圧縮記事を解凍して配信した後は、図 8 A、図 8 B、図 8 C に示するような処理例が考えられる。

まず図 7 のステップ S 3 0 5 から図 8 A のステップ S 3 1 0 に進む例では、ブログサーバ 1 は解凍記事、即ち圧縮を解凍した記事のデータ、即ち配信したウェブページデータを保存しておかずに消去する例である。

これは、当該記事への今回のアクセス要求は、あくまで例外的にアクセスが生じてしまったもので、この記事へのアクセス可能性が低いために圧縮された記事であることに代わりはないという考え方で、圧縮記事のまま保存しておくものである。

もしその後にアクセスが発生したら、その都度、解凍処理を行うことになる。アクセス要求に応じて解凍処理負担が発生するが、そもそもアクセスが少ないと考えれば、圧縮状態で保存しておくことで記憶リソース的に有利な状態を維持できる。

[0085] 一方で、圧縮記事に対するアクセスが生じたということは、その圧縮記事（アクセス可能性が低いと判定された記事）について、アクセス可能性が高まってきた可能性があると考えられることもできる。

そこで図 7 のステップ S 3 0 5 から図 8 B のステップ S 3 2 0 に進む例が考えられる。ステップ S 3 2 0 でブログサーバ 1 は、解凍した記事データをブログに組み込む。そしてステップ S 3 2 1 で圧縮記事を削除する。即ち、それまでブログに組み込んでいた圧縮記事に代えて、解凍した記事データ（解凍記事）をブログに組み込むようにする。

ステップ S 3 2 2 では、ブログサーバ 1 は管理 DB 5 3 における圧縮解凍情報を更新する。即ち当該ブログの該当の記事について、圧縮から解凍した状態の記事データ（即ち解凍記事）であることを示すように情報を更新する。また解凍の日時等の履歴情報を追加する。

[0086] このように解凍した場合、圧縮記事を解凍記事に置き換えることで、その後、アクセス要求があった場合に、解凍を行わずに配信できる。

なお、もし当該解凍記事に対して、その後もアクセスが少なく、図5の圧縮判定処理でアクセス可能性が低いと判定された場合は、図6の圧縮処理で再び圧縮されることになる。従って、相変わらず不人気な記事であった場合、再び圧縮されるため、記憶リソース維持の観点で図8Aの処理と比べて大きな不利とはならない。

また、非可逆圧縮を用いて圧縮処理を行った場合には、その後の解凍処理によって解凍した解凍記事のデータを記憶しておいたとしても、未圧縮記事を記憶しておくよりもブログDB51において占有する記憶領域が小さいため有利である。

[0087] 図8Cは、解凍記事を圧縮記事とともに保存しておく例である。図7のステップS305から図8CのステップS330に進んだ場合、ブログサーバ1は、解凍した記事データをブログに対応づけて登録しておく。但し、圧縮記事はそのまま保持する。そしてステップS331でブログサーバ1は、管理DB53における圧縮解凍情報を更新する。即ち当該ブログの該当の記事について、解凍記事が存在することを示すように情報を更新する。また解凍の日時等の履歴情報を追加する。

[0088] この場合、解凍記事を保存しておくことで、その後にアクセス要求があった場合に、解凍処理を行わずに配信できる。

但し、解凍記事と圧縮記事の両方を保存しておくことで、記憶リソースの負担を大きくする。そこで例えば解凍記事は、或る一定期間を経過したら削除することが考えられる。このようにすれば、アクセス要求が生じた場合、一定期間は、再度アクセスがあったときに解凍処理を経ずに配信できる状態を作ることができる。

また、このように圧縮記事と解凍記事を併存させた場合、当該記事がその後の図5の処理で圧縮対象となった場合、図6の圧縮処理では、解凍記事を削除するという処理を行えばよい。

[0089] <4. 第2の実施の形態>

第2の実施の形態では、取得する人気度指数が複数の情報から算出された値とされる。即ち、第1の実施の形態では、管理DB53にブログ実績情報として記憶された情報のうちの一つを人気度指数として取得するのに対し、第2の実施の形態では、管理DB53にブログ実績情報として記憶された複数の情報から人気度指数を算出して取得する。

同様に、記事ごとのアクセス可能性指数に関しても、ブログ実績情報として記憶された複数の情報からアクセス可能性指数を算出して取得する。

[0090] このために、ブログサーバ1は図5に示す圧縮判定処理を行う前に人気度指数やアクセス可能性指数を算出しておくことが考えられる。もちろん、その都度算出してもよい。

[0091] 図9は、人気度指数を算出する処理を示すものである。

図示するように、ブログサーバ1はステップS401において、処理対象のブログ即ち人気度指数の算出対象とするブログを特定する。

続いて、ブログサーバ1はステップS402において、人気度指数算出に用いるブログ実績情報を管理DB53から取得する。具体的には、ブログ実績情報としてブログごとに記憶されたブログの総ページビュー数、ブログにアクセスしたユニークユーザ数、総被リンク数、ブログのページランクを示す値、ブログに対するアクセス要求が無い期間の長さ、所定期間におけるブログ全体についての総ページビュー数、ブログ全体についてのページビューの増加傾向を示す値、ブログ更新頻度を示す値、ブログページ上に掲載される広告に対するクリック数、ブログ全体についてのデータ量の増加傾向を示す値などの情報から複数の情報を取得する。

[0092] ステップS403でブログサーバ1は、取得した複数の情報から人気度指数を算出する。人気度指数算出のための情報として管理DB53からブログの総ページビュー数及びブログにアクセスしたユニークユーザ数を取得した場合、ブログの総ページビュー数及びブログにアクセスしたユニークユーザ数それぞれについて人気度指数を算出するための元となる数値を求める。例

例えば、ブログの総ページビュー数について１（低人気）～１００（高人気）に正規化した数値（正規化値）を求め、ブログにアクセスしたユニークユーザ数について１～１００に正規化した数値（正規化値）を求めることが考えられる。即ち、ステップＳ４０３の処理では、ブログの総ページビュー数の正規化値とブログにアクセスしたユニークユーザ数の正規化値を加算する処理や平均値を算出する処理を実行する。これにより、処理対象とされたブログの人気度指数を算出する。

複数の情報から人気度指数を求めることによって、より総合的に処理対象ブログの人気度合いを推し量ることが可能となる。例えば、総ページビュー数が多くても、アクセスしたユニークユーザ数が少ないブログである場合、特定のユーザに人気のあるブログであることが推測される。従って、複数の情報を組み合わせて人気度指数を求めることは、ブログごとの適切な人気度指数を算出するために有効であることが多い。

なお、上記二つの情報を用いて人気度指数を求めることは一例であり、三つ以上の情報から算出してもよいし、全ての指標を用いて算出してもよい。

[0093] ステップＳ４０４でブログサーバ１は、算出した人気度指数を管理ＤＢ５３に記憶する。

人気度指数を管理ＤＢ５３に記憶しておくことにより、図５に示す圧縮判定処理ではステップＳ１０２の処理で記憶された人気度指数そのものを取得すればよい。

[0094] 次に、ステップＳ４０５でブログサーバ１は、次の処理対象のブログがあるか否かを判定し、次の処理対象のブログがある場合にはステップＳ４０１に戻り、次のブログを特定する。一方、次の処理対象のブログが無い場合には、ステップＳ４０５から図９に示す人気度指数算出処理を終了する。

なお、処理対象のブログはブログＤＢ５１に記憶された全てのブログを対象としてもよいし、一部のブログを対象としてもよい。一部のブログのみを対象とする場合には、ブログに対するアクセス数が急増したブログなど、管理ＤＢ５３に記憶されたブログ実績情報の変動が大きいブログを対象とする

ことが考えられる。これにより、人気度指数の再算出が必要なブログを効率よく処理対象のブログとして選択することができる。

[0095] 図10は、アクセス可能性指数を算出する処理を示すものである。

ブログサーバ1はステップS501において、圧縮対象ブログの中から一つのブログを特定する。以降の処理では、特定したブログに対してアクセス可能性指数を算出する。

ステップS502でブログサーバ1は、処理対象のブログに含まれる複数の記事ごとのアクセス可能性指数を算出するために用いる情報をブログDB51のブログ実績情報から取得する。即ち、記事が10個あるブログが処理対象であれば、10個の記事ごとに情報を取得する。

具体的には、ブログ実績情報として記事ごとに記憶された記事ごとのページビュー数、記事にアクセスしたユニークユーザ数、記事ごとの総被リンク数、記事のランクを示す値、記事に対するアクセス要求が無い期間の長さ、所定期間における記事についてのページビュー数、記事についてのページビューの増加傾向を示す値、記事ページ上に掲載される広告に対するクリック数などの情報から複数の情報を取得する。

[0096] ステップS503でブログサーバ1は、取得した複数の情報からアクセス可能性指数の記事ごとに算出する。このとき、情報の種類ごとに正規化した数値を加算することにより当該記事のアクセス可能性指数を求めてもよいし、情報の種類ごとの重要性に応じて重みを付けて算出してもよい。

続いて、ブログサーバ1はステップS504において、算出した記事ごとのアクセス可能性指数を管理DB53に記憶する。アクセス可能性指数を管理DB53に記憶しておくことにより、図5に示す圧縮判定処理におけるステップS107の処理では、記憶されたアクセス可能性指数そのものを取得すればよい。

[0097] 次に、ステップS505でブログサーバ1は、次の処理対象となる圧縮対象ブログがあるか否かを判定し、次の処理対象のブログがある場合にはステップS501に戻り、次の圧縮対象ブログを特定する。一方、次の処理対象

となる圧縮対象ブログが無い場合には、ステップS505から図10に示すアクセス可能性指数算出処理を終了する。

なお、処理対象の圧縮対象ブログは圧縮対象とされたブログ全てとしてもよいし、一部としてもよい。一部とする例としては、例えば、全ての記事を圧縮しなくてはならないほど圧縮対象ブログ判定閾値を大幅に超えているブログは、記事ごとにアクセス可能性指数を算出する必要がなくなるため、処理対象外としてもよい。換言すれば、圧縮対象ブログ判定閾値を少し超えている程度の圧縮対象ブログは、何れの記事を圧縮するべきかを検討する必要があるため、アクセス可能性指数の算出対象のブログとする。

[0098] なお、図10のアクセス可能性指数算出処理は、上述したように圧縮処理の対象となったブログのみを処理の対象としている。これは、例えば、ブログDB51に記憶された全てのブログに対して図9の処理を行うことにより人気度指数を算出した後に、図5のステップS104の処理を行いブログごとに圧縮対象ブログであるか否かを判定する。そして、圧縮対象ブログであると判定されたブログに対して図10に示すアクセス可能性指数算出処理（特にステップS502乃至S504の処理）を行うことで実現できる。

また、図5に示すステップS102の代わりに図9のステップS402及びS403を実行し、図5に示すステップS107の代わりに図10のステップS502及びS503を行うことでも実現可能である。このとき、ステップS502及びS503の処理は、ステップS105で対象とした記事に対してのみ行えばよい。

[0099] なお、図9の人気度指数算出処理及び図10のアクセス可能性指数算出処理をバッチ処理で定期的に行う場合には、人気度指数算出処理及びアクセス可能性指数算出処理の処理対象をブログDB51に記憶された全てのブログ及び全ての記事としてもよい。

例えば、図9、図10に示す処理を全てのブログ及び全ての記事に対して実行しておき、その後で、図5及び図6に示した圧縮判定処理及び圧縮処理を行うことによって、適切に圧縮処理を行うことができる。このとき、ステ

ップS501の処理は、圧縮対象ブログを特定する処理ではなく、処理対象のブログを選択する処理となる。

[0100] 人気度指数及びアクセス可能性指数を複数の情報から算出する場合においては、それぞれの指数を事前にバッチ処理などで算出しておくことにより、処理のスケジュールの自由度を高めることができる。

例えばあるブログの急激な人気上昇に対応するために、人気度指数を短い間隔で算出したい場合などに好適である。

[0101] <5. 第3の実施の形態>

第3の実施の形態では、処理対象のブログの総データ量が所定値以下となるまで、アクセス可能性指数が低い記事から圧縮をしていく例を説明する。

圧縮処理の流れを図11に示す。処理のおおまかな流れは図6に示す圧縮処理と同様であるため、説明を適宜省略する。

[0102] ブログサーバ1は、ステップS201乃至S203の処理を実行することにより、圧縮対象ブログの一つを選択する。そして、選択したブログに対して、以降の各処理を実行する。

まず、ブログサーバ1はステップS211において、選択したブログに属する各記事のうちアクセス可能性指数の低い記事を選択する。そして、当該記事に対してステップS205乃至S209の処理を行うことにより記事の圧縮等を行う。ステップS211の処理は、図6におけるステップS204の処理に代わる処理である。

[0103] 一つの記事の圧縮を終えたブログサーバ1は、続くステップS212において、当該ブログの全体の総データ量が所定値以下であるか否かを判定する。

所定位置よりも大きい場合、このブログの各記事に対する圧縮は十分ではないと判定し、ステップS211の処理に戻って次の記事を選択する。

なお、全ての記事を圧縮したために次に選択すべき記事が存在しない場合は、ステップS211の処理に遷移せずにステップS210へ遷移して、次のブログに対する処理を開始するか否かを判定してもよい。

[0104] 選択したブログの総データ量が所定値以下となった場合、ブログサーバ1は当該ブログに対する圧縮処理を終了し、次の処理対象のブログを選択するための処理としてステップS210を実行する。

[0105] ステップS212の処理で用いる所定値としては、例えば、圧縮対象ブログ判定閾値を用いることが考えられる。圧縮対象ブログ判定閾値を用いることにより、判定に用いる数値を新たに算出したり記憶したりする必要が無いため、処理負担の軽減や記憶リソースの確保に寄与することができる。

また、他にも、所定値として圧縮対象ブログ判定閾値に0.8などの係数を乗算した値を用いてもよい。圧縮対象ブログが記事を頻繁に投稿するユーザのものであった場合、これにより1回の投稿で圧縮対象ブログ判定閾値を再度超えてしまうことを抑制できる。従って、当該ブログが再び圧縮対象ブログとなるまでの時間を長くすることができるため、図11に示す圧縮処理（或いは図5のステップS105乃至S110の処理）の実行頻度を下げることができる。これにより、ブログサーバ1の処理負担の軽減を図ることができる。

[0106] 第3の実施の形態によれば、処理対象のブログの総データ量が所定値以下となったことに応じて即座に各記事の圧縮処理が終了するため、過剰となり得る圧縮処理を抑制することができ、ブログサーバ1の処理負担の軽減を図ることができる。

なお、状況に応じて所定値を設けたり設けなかったりしてもよい。例えば、処理対象のブログの全ての記事のアクセス可能性指数が低い場合などは、所定値を設けずに全ての記事に対する圧縮処理（即ち図11のステップS205～S209の処理）を行ってしまってもよい。このようなブログは、閲覧者のアクセスによる記事の解凍処理を実行する可能性が低いため、解凍処理の実行による処理負担の増加を懸念することなく記憶リソースの確保を行うことができる。

[0107] <6. 第4の実施の形態>

第4の実施の形態では、当該ブログの総データ量が圧縮対象ブログ判定閾

値をどの程度超えているかに基づいて、当該ブログに属する記事に対する圧縮可否判定を行う例について説明する。

圧縮判定処理について図12に示す。処理のおおまかな流れは図5に示す圧縮判定処理と同様であるため、説明を適宜省略する。

[0108] ブログサーバ1は、ステップS101乃至S104において、ブログが圧縮対象ブログであるか否かを判定する。このとき圧縮対象ブログとして判定されたブログに関しては、総データ量が圧縮対象ブログ判定閾値をどの程度超えたのか（即ち超過容量）を把握しておく。

続いて、ブログサーバ1は、ステップS105において選択した記事について、続くステップS106～S109を実行することにより当該記事を圧縮すべきか否かを判定する。このとき、アクセス可能性指数の大きさにより圧縮すべきか否かを判定するが、本実施の形態では、上記超過容量も加味して圧縮すべきか否かの判定を行う。

具体的に例えば、先の第1の実施の形態において、図5のステップS108の説明の際に、アクセス可能性指数として当該記事に対するアクセス要求が無い期間の長さを選択した場合、当該期間長が3年以上であるか未満であるかを判定し、3年以上アクセス要求が無い記事はアクセス可能性が低いとして圧縮対象記事と判定する例を挙げた。

[0109] 本実施の形態によれば、この「3年」という閾値を超過容量に応じて変えるということである。一例を挙げると、超過容量が多いブログ、即ち多くの記事を圧縮しなければならないブログに関しては、「2年」という閾値を設ける。これにより、2.5年間アクセス要求が無いような記事も圧縮処理の対象となるため、記憶リソースの確保を促進させることができる。また、超過容量の少ないブログ、即ち少しの記事を圧縮するだけでよいブログに関しては、「5年」という閾値を設ける。これにより、3年間アクセス要求が無い記事は圧縮処理の対象外となるため、圧縮記事に対する解凍処理が発生する可能性を低減させつつ効率的な記憶リソースの確保を行うことができる。

[0110] そのために、ブログサーバ1はステップS112においてアクセス可能性

指数を取得した後、ステップS 1 1 3においてブログの総データ量の超過容量に基づく判定閾値を取得する。ステップS 1 1 3においては、判定閾値の取得の代わりに判定閾値を算出する処理、即ち上述した2年や3年や5年などの閾値を算出する処理を実行してもよい。

[0111] 第4の実施の形態によれば、各ブログの総データ容量に応じて適切な判定閾値が設定されることにより、好適な記憶リソースの確保を行うことができる。

[0112] <7. 第5の実施の形態>

第5の実施の形態では、ブログサーバ1は、各記事について、人気度指数及びアクセス可能性指数に基づいた圧縮可否判定を実行するだけでなく、既に圧縮した記事の解凍可否判定も実行する。

[0113] ブログサーバ1は図13の圧縮解凍判定処理を例えば定期的、或いは所定のタイミングで実行する。この図13の処理は第1の実施の形態における図5の圧縮判定処理に代わる処理と考えればよい。図5と同一の処理については同一のステップ番号を付して重複説明を避ける。

[0114] ブログサーバ1は処理対象のブログを特定し(S 1 0 1)、人気度指数を取得(S 1 1 4)した後、ステップS 1 0 5、S 1 1 5、S 1 1 6、S 1 1 7、S 1 1 0で各記事について判定を行う。

ステップS 1 1 4では、図5のステップS 1 0 2のように人気度指数としてのブログ実績情報を取得してもよいし、図9で算出した人気度指数を取得してもよい。

ブログサーバ1は、ステップS 1 0 5で一つの記事を選択したら、圧縮済か否かに関わらずステップS 1 1 5に進んでアクセス可能性指数を取得する。この処理に関しても、図5のステップS 1 0 7のようにアクセス可能性指数としてのブログ実績情報を取得してもよいし、図10で算出したアクセス可能性指数を取得してもよい。

そしてブログサーバ1はステップS 1 1 6で圧縮可否及び解凍可否の判定を行う。

即ちブログサーバ1は、未圧縮記事もしくは解凍記事については、図5のステップS108と同様に、圧縮可否判定を行う。

[0115] 一方、圧縮記事についてはブログサーバ1は、人気度指数及びアクセス可能性指数を用いて解凍可否判定を行う。

例えば、圧縮可否判定の場合と同様に、総ページビューなどの情報を用いて、人気度指数を高／中／低の3段階などに分ける。

そしてアクセス可能性指数として、例えば当該記事のページビュー数Nを用い、

- ・人気度指数が低のブログの場合・・・記事のページビュー数NがN1以上であれば解凍

- ・人気度指数が中のブログの場合・・・記事のページビュー数NがN2以上であれば解凍

- ・人気度指数が高のブログの場合・・・記事のページビュー数NがN3以上であれば解凍

とする。但し $N1 > N2 > N3$ である。

つまり人気ブログの圧縮記事であれば、ページビュー実績が少々高くなっていたら解凍可とするが、不人気ブログの圧縮記事であれば、ページビュー数により解凍可とする閾値を上昇させて判定を行う。人気ブログの記事ほど、一旦圧縮されても解凍可とされやすいことになる。これは人気ブログほど、アクセス可能性は上昇しやすいと考えられることによる。

[0116] 他の例として、アクセス可能性指数として、例えば当該記事のページビューの増加傾向を示す値Kを用い、

- ・人気度指数が低のブログの場合・・・増加傾向値KがK1以上であれば解凍

- ・人気度指数が中のブログの場合・・・増加傾向値KがK2以上であれば解凍

- ・人気度指数が高のブログの場合・・・増加傾向値KがK3以上であれば解凍

とすることも考えられる。但し $K1 > K2 > K3$ である。

つまり人気ブログの圧縮記事であれば、ページビューの少々の増加傾向が観測されたら解凍可とするが、不人気ブログの圧縮記事であれば、ページビューの大幅な増加傾向が観測されたら解凍可するというような判定を行う。この場合も、人気ブログの記事ほど、一旦圧縮されても解凍可とされやすいことになる。

[0117] そしてステップ S 1 1 7 ではブログサーバ 1 は、当該記事についての圧縮可否判定または解凍可否判定を判定情報として記憶する。例えば管理 DB 5 3 の判定情報として当該記事について圧縮可否のフラグまたは解凍可否のフラグを更新または維持する。

ステップ S 1 1 0, S 1 1 1 は図 5 と同様である。

[0118] ブログサーバ 1 はまた、図 1 4 の圧縮解凍処理を逐次実行する。これは第 1 の実施の形態における図 6 の圧縮処理に代わるものである。図 6 と同一処理は同一のステップ番号を付し、詳細な説明は避ける。

[0119] ブログサーバ 1 はステップ S 2 0 1 で処理対象のブログを特定し、ステップ S 2 0 2 で判定情報を取得する。

ブログサーバ 1 はステップ S 2 1 3 で、圧縮可否の判定情報を参照して圧縮する記事があるかどうかを判定する。そしてステップ S 2 0 4 で圧縮する記事を特定する。ステップ S 2 1 3 では、圧縮記事があるか否かを判定してもよいが、処理対象のブログが圧縮対象ブログであるか否かを判定してもよい。

1 または複数の記事が圧縮する記事として特定された場合、ブログサーバ 1 はステップ S 2 0 5 で、1 または複数の各記事データの圧縮処理を行い、図 5 の場合と同様にステップ S 2 0 7, S 2 0 8, S 2 0 9 で、圧縮した記事についてのタグ設定、タグ登録、及び元の記事データの削除を行う。

[0120] 以上のステップ S 2 0 5 ~ S 2 0 9 を行った後、もしくはステップ S 2 1 3 で圧縮する記事が存在しないとされた後、ブログサーバ 1 はステップ S 2 1 4 で、解凍する記事があるか否かを判定する。即ちステップ S 2 0 2 で取

得した判定情報を参照して、圧縮されている記事の中で解凍すべき記事があるか否かを判定する。解凍する記事が存在しなければステップS 2 1 8に進む。

[0121] 1 または複数の解凍すべき記事が存在する場合、ブログサーバ1はステップS 2 1 5で解凍する記事の特定を行い、続くステップS 2 1 6で、特定した記事の解凍処理を行う。この場合、解凍した記事データは、それまでの圧縮記事に代えてブログに組み込むようにする。またステップS 2 1 7でブログサーバ1は圧縮記事を削除する。

これにより、ブログ内の圧縮されていた記事が、アクセス可能性の上昇に応じて、解凍記事に回復されたことになる。

[0122] ブログサーバ1はステップS 2 1 8で管理DB 5 3における圧縮解凍情報を更新する。ここでは当該ブログ内の圧縮を行った記事について、圧縮状態であることを示すようにフラグ情報を更新する。また圧縮履歴を追加する。更に当該ブログ内の解凍を行った記事について、解凍状態であることを示すようにフラグ情報を更新する。また解凍履歴を追加する。

[0123] 以上の処理により一つのブログについての圧縮処理及び解凍処理を終えたら、ブログサーバ1はステップS 2 1 0で他のブログについて処理を行うか否かを確認する。

そして今回処理対象とする全てのブログについて処理を終えていた場合は、ステップS 2 1 0から図1 4の圧縮解凍処理を終える。

以上の図1 4の圧縮解凍処理を行うことで、図1 3の圧縮解凍判定処理で圧縮可と判定された記事についての実際の圧縮処理や、解凍可と判定された記事についての実際の解凍処理が行われる。これにより記憶リソースの回復が図られるとともに、アクセス可能性が高くなった記事については解凍が行われ、アクセス時のパフォーマンスを向上させる。

[0124] 例えば或るブログの記事について、そのブログの記述者によって内容の変更があったり、その記事に対して閲覧者からコメントが投稿されたりした場合など、その記事自体に更新があった場合には、その記事に対するアクセス

可能性は高くなる。本実施の形態によれば、このような記事は、圧縮を解いておくという動作が実現される。

そこで、このような動きを事前に察知して、既に圧縮したもののアクセス可能性の高まったブログ記事については、圧縮状態を解凍しておく。

このようにすれば、実際にアクセスしてきた場合に、その時点で解凍処理をせずにすぐに記事を配信することができる。

[0125] < 8. 第6の実施の形態 >

第6の実施の形態として、ブログサーバ1が記事内容に基づいて、既に圧縮した記事を解凍するか否かを判定する例を説明する。

例えば、あるテーマについて言及した記事について、当初にアップロードされた時点ではあまり話題にならなかったものの、数年後に、その記事で扱ったテーマに関する何らかの事件や事象が世の中で起きたために、その記事が掘り起こされる可能性が高まる（つまり、アクセス可能性が高まる）ことや、その記事に対する被リンク数が増加するといったことが起きうる。

そこで、このような動きを事前に察知して、既に圧縮したもののアクセス可能性の高まった記事については、圧縮状態を解凍しておくようにする。

このようにすれば、実際にアクセス要求があった場合に、その時点で解凍処理をせずにすぐに記事を配信することができる。

[0126] 図15に解凍判定処理を示す。ブログサーバ1は、第1の実施の形態の各処理に加え、例えば定期的などに逐次この解凍判定処理を行う。

ステップS601でブログサーバ1は1または複数の抽出語句を設定する。例えば時事語を抽出語句とする。例えば新聞やニュースで頻出する言葉や、芸能関係でよく使われ出した言葉、流行語などである。或いは、話題になっているジャンルや関連語などを抽出語句としてもよい。例えば、オリンピック開催期間であれば、「スポーツ」というジャンルや、各種競技の名称、選手名などを抽出語句とする。

[0127] ブログサーバ1はステップS602で処理対象のブログを一つ特定する。そしてステップS603で、特定したブログに圧縮記事が存在するか否かを

確認する。例えば管理DB53の圧縮解凍情報を確認すればよい。

もし圧縮記事が存在しなければ、解凍判定の必要はないためステップS607に進む。

圧縮記事が存在するブログであった場合は、ブログサーバ1はステップS604に進み、当該ブログの圧縮記事タグの情報を管理DB53から取得する。圧縮記事タグとしては、少なくとも現在圧縮状態である1または複数の圧縮記事について設定されたタグ情報を含む。即ち、図6のステップS208などで登録されたタグ情報である。

[0128] ステップS605でブログサーバ1は、ステップS601で設定した抽出語句と管理DB53から取得した圧縮タグ情報を比較し、解凍する記事を判定する。

つまり抽出語句と同一または類似の圧縮記事タグが登録されている記事を抽出し、解凍する記事と判定する。

ステップS606でブログサーバ1は、判定情報を管理DB53に記憶する。つまり解凍すると判定した記事の情報を記憶する。

[0129] 以上の処理により一つのブログについての解凍判定処理を終えたら、ブログサーバ1はステップS607で次の処理対象のブログがあるか否かを確認する。ある場合はステップS602に戻り、次の処理対象のブログを特定して上記同様のステップS603以降の処理を行う。

今回処理対象とする全てのブログについて処理を終えていた場合は、ステップS607から図15の解凍判定処理を終える。

[0130] この解凍判定処理とともに、ブログサーバ1は図16の解凍処理を逐次実行する。

ブログサーバ1はステップS201で処理対象のブログを特定し、ステップS202で判定情報を取得する。

ブログサーバ1はステップS214Aで、取得した判定情報を参照して、圧縮されている記事のうちで解凍すべき記事があるか否かを判定する。解凍すべき記事が存在しなければステップS210に進む。

[0131] 1または複数の解凍すべき記事が存在すると判定した場合、ブログサーバ1はステップS215Aで、解凍する記事を特定した後、ステップS216Aで当該特定された1または複数の記事の解凍処理を行う。この場合、解凍した記事データは、それまでの圧縮記事に代えてブログに組み込むようにする。またステップS217Aでブログサーバ1は圧縮記事を削除する。これにより、ブログ内の圧縮されていた記事が解凍記事に回復されたことになる。

ブログサーバ1はステップS218Aで管理DB53における圧縮解凍情報を更新する。ここでは当該ブログ内の解凍を行った記事について、解凍状態であることを示すようにフラグ情報を更新する。また解凍履歴を追加する。

[0132] 以上の処理により一つのブログについての解凍処理を終えたら、ブログサーバ1はステップS210で次の処理対象のブログがあるか否かを確認する。ある場合はステップS201に戻って、他の処理対象のブログを特定し、同様の処理を実行する。

今回処理対象とする全てのブログについて処理を終えていた場合は、ステップS210から図16の解凍処理を終える。

[0133] 以上の図16の処理により、図15の解凍判定処理で解凍可と判定された記事についての実際の解凍処理が行われる。

これにより、時事語、流行語、流行のテーマなどを含む記事が、アクセス可能性が高くなるであろうと予測して解凍が行われる。これによりその後のアクセス増加時に解凍処理を不要にすることができる。

[0134] <9. 第7の実施の形態>

第7の実施の形態は、ブログの人気度指数の変化を監視し、或るブログについて人気度指数の上昇傾向を検知した場合に、該ブログに含まれる圧縮済の記事を解凍する記事と判定する例である。つまり、人気上昇したブログについては、個々の記事についての判断をせずに、そのブログに含まれる圧縮記事は解凍可とする。

[0135] この場合の解凍判定処理を図 17 に示す。

ブログサーバ 1 はステップ S 602 で処理対象のブログを一つ特定する。そしてステップ S 603 で、管理 DB 53 の圧縮解凍情報を参照し、特定したブログに圧縮記事が存在するか否かを確認する。

もし圧縮記事が存在しなければ、解凍判定の必要はないためステップ S 607 に進む。

[0136] 圧縮記事が存在するブログであった場合は、ブログサーバ 1 はステップ S 608 に進み、当該ブログのブログ実績情報を管理 DB 53 から取得する。

ここで取得するブログ実績情報は、処理対象のブログの人気度指数を示す情報である。例えば、ブログ実績情報として記憶された情報から人気度指数として採用する情報を一つ取得してもよいし、複数の情報から算出された人気度指数の情報を取得してもよい。なお、この処理では人気度指数を取得する代わりに複数の情報から人気度指数を算出してもよい。

ステップ S 608 で取得する人気度指数を示す情報は、期間ごとの値として参照できる情報が望ましい。つまり人気の変動を判定できる情報である。

例えば当該ブログの総ページビュー数やアクセスしたユニークユーザ数、総被リンク数、総コメント数などであれば、期間ごと、例えば 1 日ごとや 1 週間ごとの値であることが好適である。

また、ブログ全体についてのページビューの増加傾向を示す値、ブログ全体についてのデータ量の増加傾向を示す値を取得してもよい。これらの値はそのまま人気の変動を表す指標となるため、ステップ S 422 で取得する情報として好適である。

[0137] ステップ S 609 でブログサーバ 1 は、上記の人気度指数から、当該ブログの人気の傾向を判定する。例えば「人気上昇傾向」「人気変動なし」「人気下降傾向」などを判定する。例えば総ページビュー数やユニークユーザ数の期間ごとの値やページビューの増加傾向や減少傾向から判定できる。少なくとも、現在人気上昇傾向にあるか否かの判定のみでもよい。

そしてブログサーバ 1 は、「人気変動なし」または「人気下降傾向」と判

定した場合はステップS 6 1 0からS 6 0 7に進み、「人気上昇傾向」と判定した場合はステップS 6 1 1に進む。

[0138] ステップS 6 1 1でブログサーバ1は、当該ブログに含まれる1または複数の圧縮記事を、全て解凍可と判定し、その判定情報、つまり解凍すると判定した記事の情報を管理DB 5 3に記憶する。

[0139] 以上の処理により一つのブログについての解凍判定処理を終えたら、ブログサーバ1はステップS 6 0 7で次の処理対象のブログがあるか否かを確認する。ある場合はステップS 6 0 2に戻り、次の処理対象のブロックを特定して上記同様のステップS 6 0 3以降の処理を行う。

今回処理対象とする全てのブログについて処理を終えていた場合は、ステップS 6 0 7から図1 7の解凍判定処理を終える。

[0140] この解凍判定処理とともに、ブログサーバ1は図1 6の解凍処理または図1 4の圧縮解凍処理を逐次実行することで、人気上昇傾向のブログにおける圧縮記事は解凍されることになる。

人気上昇したブログについては、どの記事もアクセス可能性が高くなるため、圧縮記事については予め解凍しておくことで、閲覧時の解凍処理を不要とし、応答性の向上やその際の処理負担の削減を実現できる。

[0141] なお、図1 7のステップS 6 1 1の時点で、当該ブログに含まれる圧縮記事の解凍を実行するようにしてもよい。

[0142] <1 0. 第8の実施の形態>

第8の実施の形態は、圧縮記事についてアクセスに応じて解凍が行われた場合、ブログサーバ1は解凍から所定期間、当該記事を圧縮する記事と判定しないようにする例である。例えば図8 Bまたは図8 Cで説明したようにアクセスに応じて解凍が行われた後、その解凍記事を記憶しておく場合に、図5の処理に代えて用いることができる圧縮判定処理である。

[0143] 図1 8に第8の実施の形態としての圧縮判定処理を示す。図5と同一の処理については同一のステップ番号を付して重複説明を避ける。

[0144] 図1 8に示す圧縮判定処理でも、ブログサーバ1は、ステップS 1 0 1で

ブログを特定し、ステップS 1 0 2で人気度指数としてのブログ実績情報を取得した後、ステップS 1 0 3で人気度指数に応じた圧縮対象ブログ判定閾値を設定したら、ステップS 1 0 4で処理対象のブログにおける総データ量が圧縮対象ブログ判定閾値を超えているか否かを判定することにより、当該ブログが圧縮対象ブログであるか否かを確認する。

[0145] 続いてブログサーバ1は、ステップS 1 0 5で記事を一つ選択し、ステップS 1 0 6で当該記事が圧縮済みであるか否かを判定する。記事が圧縮済みである場合は、ステップS 1 1 0の処理へ遷移する。

当該記事が圧縮済みでない場合、ブログサーバ1はステップS 1 1 8で当該記事が解凍された記事であるか否かを判定する。解凍された記事ではない場合、即ち未圧縮状態の記事である場合には、ブログサーバ1はステップS 1 0 7乃至S 1 0 9の各処理を当該記事に対して行う。

当該記事が解凍された記事である場合、ブログサーバ1はステップS 1 1 9において解凍から所定時間経過しているか否かを判定する。例えば、当該記事の解凍日時を確認し、現時点で所定期間を経過しているか否かを確認する。所定期間は例えば1ヶ月などとする。解凍日時の確認は管理DB53の圧縮解凍情報における履歴情報を参照して行う。なお、圧縮／解凍が複数回行われている記事の場合は、最新の解凍日時を確認する。

そして解凍から所定時間が経過している場合、ブログサーバ1は当該記事を再圧縮可能と判断し、ステップS 1 0 7乃至S 1 0 9の各処理を当該記事に対して行う。即ち、アクセス可能性指数に応じた圧縮可否判定を行い、判定情報を記憶する。

一方、解凍から所定時間経過していない場合には、ブログサーバ1は当該記事の圧縮可否判定は行わずに、即ち圧縮は行わないとされて、ステップS 1 1 0の処理に進む。

[0146] 従って、一旦圧縮した後に解凍した解凍記事については、解凍から所定期間経過するまでは、アクセス可能性の判定は行われず、圧縮可とは判定されないことになる。

このため解凍してから所定期間の間は、アクセス要求があった場合に、必ず解凍処理を行うことなく配信が可能となる。

[0147] なおステップS 1 1 8の処理は、図7のようにアクセス要求の受信時に解凍された解凍記事であるか否かを判定することを前提としているが、アクセス要求の受信としては、ユーザ端末5からの閲覧のためのアクセス要求を受信する以外にも、例えばクローラによるアクセス要求を受信することもあり得る。ここでいうクローラとは、各種のウェブサイトにおけるテキストデータや画像データなどを周期的に取得して自動的にDB化するプログラム（或いはそのプログラムに基づく動作を行う情報処理装置）のことである。このクローラによるアクセス要求は人気度や閲覧のためのアクセス可能性指数の変化には関与しない。

そこでクローラのアクセス要求に応じて解凍したような場合は、ステップS 1 1 8で解凍記事と判断しないようにすることが望ましい。つまりその場合は、解凍直後であっても圧縮可否判断の対象とする。

或いはクローラによるアクセス要求に応じて解凍したような場合は、図8Aのように解凍記事を保存しないようにすることも考えられる。

更には、クローラからのアクセス要求を受信した場合、圧縮記事をそもそも解凍しないようにするという手法も考えられる。

ここではクローラを例に挙げたが、一般の閲覧者たるユーザの意思と考えられるアクセス以外のアクセスは、以上のクローラの場合と同様の対応をとるとよい。

[0148] <1 1. まとめ及び変形例>

以上の実施の形態によれば次のような効果が得られる。第1～第8の実施の形態の情報処理装置としてのブログサーバ1は、1または複数の記事が含まれるブログの人気度指数を取得する人気度指数取得部12と、ブログごとの人気度指数に応じて該ブログごとに圧縮するか否かを判定するための閾値（圧縮対象ブログ判定閾値）を設定する閾値設定部13と、一つのブログに含まれる記事の総データ量と閾値とに基づいて当該一つのブログに含まれる

記事の少なくとも一部を圧縮する圧縮対象ブログとするか否かを判定し、圧縮対象ブログに含まれる各記事の中から圧縮対象とする記事を圧縮対象記事として判定する判定部14と、を備えている。

これらの機能により図5、図12、図13、図18の処理で圧縮可否の判定を行うようにしている。

ブログの記憶リソースが圧迫されてきた場合に、ブログごとに一定の閾値を設けて記事の圧縮を行うことが考えられる。本構成によれば、一律同じ閾値を設けるのではなく、ブログの人気度を示す指標（人気度指数）に基づいて決定された閾値によって圧縮対象のブログとするか否かを決定することにより、適切なブログを圧縮対象とすることができる。

人気の高いブログに属する記事は、アクセス要求がされやすいことが想定される。つまり、人気ブログの中の各記事は、不人気ブログの各記事よりもアクセス可能性は高いと言える。本実施の形態の処理によれば、ブログの人気を考慮した判定ができ、これにより圧縮対象を適切に選択できる。

また、圧縮や、圧縮に対する解凍は、処理負荷が高いが、本実施の形態の場合、圧縮や解凍をなるべく行わないようにできる。

もし、圧縮した記事に対するアクセス要求があった場合、ブログサーバ1はその記事を解凍して配信することが考えられるが、記憶リソースのために圧縮する記事は、アクセスされる可能性が低いブログに属する記事となるため、アクセス時に解凍処理が必要となる事態を極力少なくできる。これによってブログ提供のためのブログサーバ1の処理負担を軽減できる。

またアクセス可能性が低い記事を選択して圧縮することで、圧縮処理の負担も軽減される。

以上から、本実施の形態によれば、記事の圧縮によって記憶リソースの圧迫を抑えるとともに、なるべく圧縮や解凍の処理機会が少なくなるように、圧縮する記事を適切に選択することができ、サーバの処理負担の低減や、閲覧時のパフォーマンス向上を図ることができる。

[0149] 第1～第8の実施の形態のブログサーバ1は、圧縮対象ブログに属する記

事のうち圧縮対象記事の圧縮を行い、既に圧縮した記事に対するアクセス要求が発生した場合に該記事の解凍を行う圧縮解凍部 15 を備えている。

即ち圧縮解凍部 15 の機能により、図 6、図 11、図 14 に示す処理で判定部 14 の判定に沿って適切に選択された記事の圧縮を行う。またアクセスされる可能性が小さいとして圧縮された記事であっても、アクセス要求が発生することは当然あり得るが、そのような場合、図 7 の処理で解凍を行うことで、アクセス要求をしたユーザに対して記事を適切に提供する。

これにより記憶リソースの圧迫防止や処理負担の削減のための適切な圧縮・解凍を行うことができる。

[0150] 第 1 ～ 第 8 の実施の形態のブログサーバ 1 の判定部 14 は、圧縮対象ブログに含まれる記事ごとのアクセス可能性の度合いに応じて、即ちアクセス可能性指数に応じて、記事ごとに圧縮対象記事とするか否かを判定する。

アクセス可能性が低い記事を圧縮することでサーバの記憶リソースの圧迫を回避できる。これらの判定処理では、ブログに含まれる記事のうちで、アクセスされる可能性の小さい記事（例えば不人気記事）を圧縮するために、アクセスされる可能性が小さい記事であるか否かを判定する。

単にブログの人気度を示す指標（人気度指数）に基づいて記事を圧縮するのではなく、アクセス可能性が小さいということを示す指標（アクセス可能性指数）を用いて圧縮対象のブログを決定することにより、真にアクセスされる可能性が低い記事が圧縮対象とされる。人気の高いブログの記事の場合、その記事自体の実績からアクセス可能性が低いとされても、他の記事との関連などによりアクセスされることも想定される。つまり、人気ブログの中の不人気記事は、不人気ブログの不人気記事よりもアクセス可能性は高いと言える。本実施の形態の処理によれば、ブログの人気だけでなく記事に対するアクセス可能性を考慮した判定ができ、これにより圧縮対象を適切に選択できる。

また、圧縮や、圧縮に対する解凍は、処理負荷が高いが、本実施の形態の場合、圧縮や解凍をなるべく行わないようにできる。

もし、圧縮した記事に対するアクセス要求があった場合、ブログサーバ1はその記事を解凍して配信することが考えられるが、記憶リソースのために圧縮する記事は、真にアクセス可能性が低い記事となるため、アクセス時に解凍処理が必要となる事態を極力少なくできる。これによってブログ提供のためのブログサーバ1の処理負担を軽減できる。

また真にアクセス可能性が低い記事を選択して圧縮することで、圧縮処理の負担も軽減される。

以上から、本構成によれば、記事の圧縮によって記憶リソースの圧迫を抑えたとともに、なるべく圧縮や解凍の処理機会が少なくなるように、ブログ内で圧縮する記事を適切に選択することができ、サーバの処理負担の低減や、閲覧時のパフォーマンス向上を図ることができる。

[0151] 第8の実施の形態のブログサーバ1の判定部14は、アクセス要求に応じて既に圧縮した記事の解凍が行われた場合、解凍から所定期間、当該記事を圧縮対象記事と判定しない。

第8の実施の形態では、圧縮された記事についてアクセスに応じて解凍が行われた場合、解凍から所定期間、当該記事を圧縮する記事と判定しないようにしている（図18参照）。

圧縮した後にアクセスがあって解凍して配信した場合、その記事についてはアクセス要求がなされる可能性が高まったと考えることができる。そこで、解凍したままとし、次のアクセス要求を受信した際に解凍処理を不要とする。

但し、アクセスが単発的なものであって、引き続き不人気記事であることもある。そこで所定期間経過後は、ブログの人気度指数や記事ごとのアクセス可能性指数に基づいて適宜圧縮を行えばよい。これにより、解凍したままとしておくことが無駄な記憶リソース消費となっている場合に対処でき、記憶に必要な容量を低減させ、記憶リソースの圧迫を回避することができる。

[0152] 第7の実施の形態のブログサーバ1の判定部14は、既に圧縮した記事について、同一の圧縮対象ブログの他の記事に対するページビューの増加傾向

を示す値に応じた解凍の是非を判定し、圧縮解凍部 15 は解凍の是非に応じて当該記事を解凍しておく。

第 7 の実施の形態では、ブログの人気度指数の変化を監視し、或るブログについて人気度指数の上昇傾向を検知した場合に、該ブログに含まれる圧縮済の記事を解凍する記事と判定するようにしている（図 17 参照）。

或るブログが何らかの人気記事によってアクセス数が顕著に上昇したような場合、そのブログに含まれる別の記事は、これまでアクセスされていなかったとしても、今後アクセスされる可能性が高まる。そこで当該ブログ内の全ての圧縮記事を解凍する記事と判定する。

このようにすれば、実際にアクセスがあった際に、その時点での解凍処理は不要となり、レスポンスよく記事を配信することができる。

[0153] なお第 7 の実施の形態の図 17 の処理の変形例として、人気上昇と判定されたブログにおける全ての圧縮記事ではなく、一部の圧縮記事を解凍する記事と判定してもよい。例えば投稿日時が現在から過去に所定期間内の記事を解凍する記事とすることで、むやみに多数の記事を解凍せず、記憶リソースの維持に好適である。例えば長期にわたって多数の記事が投稿されており、比較的古い記事の多数が圧縮されているようなブログについては、全部ではなく一部の圧縮記事を解凍対象とすることが好ましい。

或いは、圧縮記事の数が所定数以上の場合は、一部（例えば半分）のみを解凍対象とすることも考えられる。

また解凍する記事数の上限を決め、上限数以上の圧縮記事が存在する場合は、投稿日時の新しい記事から上限数の記事を解凍する記事と判定してもよい。

[0154] 第 6 の実施の形態のブログサーバ 1 の判定部 14 は、記事内容に基づいて、既に圧縮した記事を解凍するか否かを判定する。

第 6 の実施の形態では、記事内容に基づいて、既に圧縮した記事を解凍するか否かを判定するようにしている（図 15 参照）。

例えば圧縮した記事の内容として、或る設定したキーワードや時事語を含

む記事、特定のテーマの記事などを抽出し、これらを解凍対象とする。

例えば、あるテーマについて言及したブログ記事について、当初にアップロードされた時点ではあまり話題にならなかったものの、数年後に、その記事で扱ったテーマに関する何らかの事件や事象が世の中で起きたために、その記事が掘り起こされる可能性が高まる（つまり、アクセス可能性が高まる）ことや、その記事に対する被リンク数が増加するといったことが起きうる。

そこで、このような動きを記事内容、具体的には時事語、設定したキーワード、記事のテーマ等により事前に察知して、既に圧縮したもののアクセス可能性の高まった記事については、圧縮状態を解凍しておく。

このようにすれば、実際にアクセスがあった際に、その時点での解凍処理を不要とし、すぐに記事を配信することができる。

[0155] なお、圧縮した記事については、内容を検索することが困難にもなる。そこで、管理DB53に圧縮記事タグとしてキーワードやテーマ等を示すタグ情報を記憶しておくようにしている。これにより、解凍判定を容易かつ適切に行うことができる。

またタグ設定及び管理DB53へのタグ登録は、圧縮時に行っている（図6，図11，図14参照）。これによりタグ設定及び登録の処理の実行を必要最小限にでき、処理負担の増大を抑止し、またDBにおける記憶リソースをむやみに圧迫しないようにできる。

[0156] 上記した実施の形態の処理例は一例であり、他にも各種の変形例が想定される。

ブログの人気度指数について例えば3段階以上の多段階に分けた場合、最も人气が低いランクであると判定されたブログについては、全記事を圧縮可と判定するようなことも考えられる。つまり人気度の n 段階の第1レベル（人气大）から第 $(n-1)$ レベル（人气低）については人気度指数及びアクセス可能性指数を用いて記事ごとの圧縮可否判定を行うが、第 n レベル（人气最低）のブログは、個々の記事を判定することなく全記事を圧縮可とする

ような例である。

特に閲覧者のアクセス（閲覧やコメント書き込み）が無いだけでなく、記述者の更新も長期にわたって無いようなブログは、実質、使用されていないブログとも考えられる。そのようなブログを、上記の人気最低のブログと判定するとよい。

これによりブログサーバ１の処理効率の向上、処理負担の削減、記憶リソースの確保を促進できる。

[0157] 圧縮処理については段階的に複数回行うようにしてもよい。

例えば図１３のように圧縮記事についても引き続きアクセス可能性の判定を行うようにする。そして圧縮済であるが、更に所定期間、アクセスがない記事は、より高い圧縮率で圧縮するなどである。

例えば１回目の圧縮では圧縮率２０％の圧縮、２回目は圧縮率５０％の圧縮、３回目は圧縮率８０％の圧縮というようにする。

また、１回目は可逆圧縮、２回目は非可逆圧縮を行うというような例もある。

また、１回目は記事の一部圧縮、２回目は記事の全体圧縮としてもよい。

また、１回目は記事のテキストのみ圧縮、２回目は加えて画像も圧縮としてもよい。

また、１回目は記事の画像のみ圧縮、２回目は加えてテキストも圧縮としてもよい。

[0158] 圧縮判定処理の対象について、一部のブログを除外することも考えられる。

例えば長期にわたって高い人気と判定されるブログについては、そのブログに含まれる記事については圧縮判定の対象外としてしまうことが考えられる。これにより図５，図１２，図１３，図１８等の処理の対象とするブログの数を削減でき、処理を効率化できる。

同様に、記事全体を圧縮済のもの、特に上述のように非常に不人気（実質的に使用されていない）と判定されるブログは、解凍判定の対象外としてし

まうことで、図 13、図 15、図 17 等の処理の対象とするブログの数を削減でき効率化が図られる。

[0159] なお、実施の形態はいわゆるブログとブログに含まれる記事を対象とし、記事の圧縮を行う処理について説明したが、このような技術は、ファイルシステムにおけるフォルダと、フォルダに含まれるファイルについても適用できる。

つまりフォルダの人気度指数は、頻繁に使用されるフォルダであるか否かという点で数値化し、それを反映させて、ファイルのアクセス可能性を基準にファイルの圧縮可否を判断するものである。

[0160] またブログは、いわゆるクラウドストレージとして実現されるシステムであってもよい。

[0161] < 12. プログラム及び記憶媒体 >

実施の形態のプログラムは、ブログサーバ 1 における少なくとも人気度指数取得部 12、閾値設定部 13、判定部 14 の処理を情報処理装置（CPU 等）に実行させるプログラムである。

[0162] 実施の形態のプログラムは、1 または複数の記事が含まれるブログの人気度指数を取得する人気度指数取得機能と、ブログごとの人気度指数に応じて該ブログごとに圧縮するか否かを判定するための閾値を設定する閾値設定機能と、一つのブログに含まれる記事の総データ量と閾値とに基づいて当該一つのブログに含まれる記事の少なくとも一部を圧縮する圧縮対象ブログとするか否かを判定し、圧縮対象ブログに含まれる各記事の中から圧縮対象とする記事を圧縮対象記事として判定する判定機能と、を情報処理装置に実行させるプログラムである。

即ちこのプログラムは、情報処理装置に対して図 5、図 12、図 13、図 18 等で説明した処理を実行させるプログラムである。

[0163] このようなプログラムにより、上述したブログサーバ 1 としての 1 または複数の情報処理装置を実現できる。

そしてこのようなプログラムはコンピュータ装置等の機器に内蔵されてい

る記憶媒体としてのHDDや、CPUを有するマイクロコンピュータ内のROM等に予め記憶しておくことができる。あるいはまた、半導体メモリ、メモリカード、光ディスク、光磁気ディスク、磁気ディスクなどのリムーバブル記憶媒体に、一時的あるいは永続的に格納（記憶）しておくことができる。またこのようなリムーバブル記憶媒体は、いわゆるパッケージソフトウェアとして提供することができる。

また、このようなプログラムは、リムーバブル記憶媒体からパーソナルコンピュータ等にインストールする他、ダウンロードサイトから、LAN、インターネットなどのネットワークを介してダウンロードすることもできる。

符号の説明

[0164] 1 ブログサーバ、2 ネットワーク、5 ユーザ端末、11 ブログ管理部、12 人気度指数取得部、13 閾値設定部、14 判定部、15 圧縮解凍部、51 ブログDB、52 画像DB、53 管理DB

請求の範囲

- [請求項1] 1 または複数の記事が含まれるブログの人気度指数を取得する人気度指数取得部と、
- 前記ブログごとの前記人気度指数に応じて該ブログごとに圧縮するか否かを判定するための閾値を設定する閾値設定部と、
- 一つの前記ブログに含まれる記事の総データ量と前記閾値とに基づいて当該一つのブログに含まれる記事の少なくとも一部を圧縮する圧縮対象ブログとするか否かを判定し、前記圧縮対象ブログに含まれる各記事の中から圧縮対象とする記事を圧縮対象記事として判定する判定部と、を備えた
- 情報処理装置。
- [請求項2] 前記圧縮対象ブログに属する記事のうち前記圧縮対象記事の圧縮を行い、既に圧縮した記事に対するアクセス要求が発生した場合に該記事の解凍を行う圧縮解凍部を備えた
- 請求項 1 に記載の情報処理装置。
- [請求項3] 前記判定部は、前記圧縮対象ブログに含まれる記事ごとのアクセス可能性の度合いに応じて記事ごとに前記圧縮対象記事とするか否かを判定する
- 請求項 1 に記載の情報処理装置。
- [請求項4] アクセス要求に応じて既に圧縮した記事の解凍が行われた場合、前記判定部は、解凍から所定期間、当該記事を前記圧縮対象記事と判定しない
- 請求項 2 に記載の情報処理装置。
- [請求項5] 前記判定部は、既に圧縮した記事について、同一の前記圧縮対象ブログの他の記事に対するページビューの増加傾向を示す値に応じた解凍の是非を判定し、
- 前記圧縮解凍部は前記解凍の是非に応じて当該記事を解凍しておく
- 請求項 2 に記載の情報処理装置。

[請求項6] 前記判定部は、記事内容に基づいて、既に圧縮した記事を解凍するか否かを判定する

請求項1に記載の情報処理装置。

[請求項7] 前記人気度指数は、ブログ全体についての総ページビュー数、記事ごとのページビュー数、ブログにアクセスしたユニークユーザ数、ブログに設定された総被リンク数、ブログに投稿された総コメント数、ブログにコメントを投稿したユニークユーザ数、ブログのページランクを示す値、ブログに対するアクセス要求が無い期間の長さ、所定期間におけるブログ全体についての総ページビュー数、ブログ全体についてのページビューの増加傾向を示す値、ブログ更新頻度を示す値、ブログページ上に掲載される広告に対するクリック数、ブログ全体についてのデータ量の増加傾向を示す値、の少なくとも一部に基づいて求められた値である

請求項1に記載の情報処理装置。

[請求項8] 1または複数の記事が含まれるブログの人気度指数を取得する人気度指数取得ステップと、

前記ブログごとの前記人気度指数に応じて該ブログごとに圧縮するか否かを判定するための閾値を設定する閾値設定ステップと、

一つの前記ブログに含まれる記事の総データ量と前記閾値とに基づいて当該一つのブログに含まれる記事の少なくとも一部を圧縮する圧縮対象ブログとするか否かを判定し、前記圧縮対象ブログに含まれる各記事の中から圧縮対象とする記事を圧縮対象記事として判定する判定ステップと、を

情報処理装置が実行する情報処理方法。

[請求項9] 1または複数の記事が含まれるブログの人気度指数を取得する人気度指数取得機能と、

前記ブログごとの前記人気度指数に応じて該ブログごとに圧縮するか否かを判定するための閾値を設定する閾値設定機能と、

一つの前記ブログに含まれる記事の総データ量と前記閾値とに基づいて当該一つの前記ブログに含まれる記事の少なくとも一部を圧縮する圧縮対象ブログとするか否かを判定し、前記圧縮対象ブログに含まれる各記事の中から圧縮対象とする記事を圧縮対象記事として判定する判定機能と、を

情報処理装置に実行させるプログラム。

[請求項10]

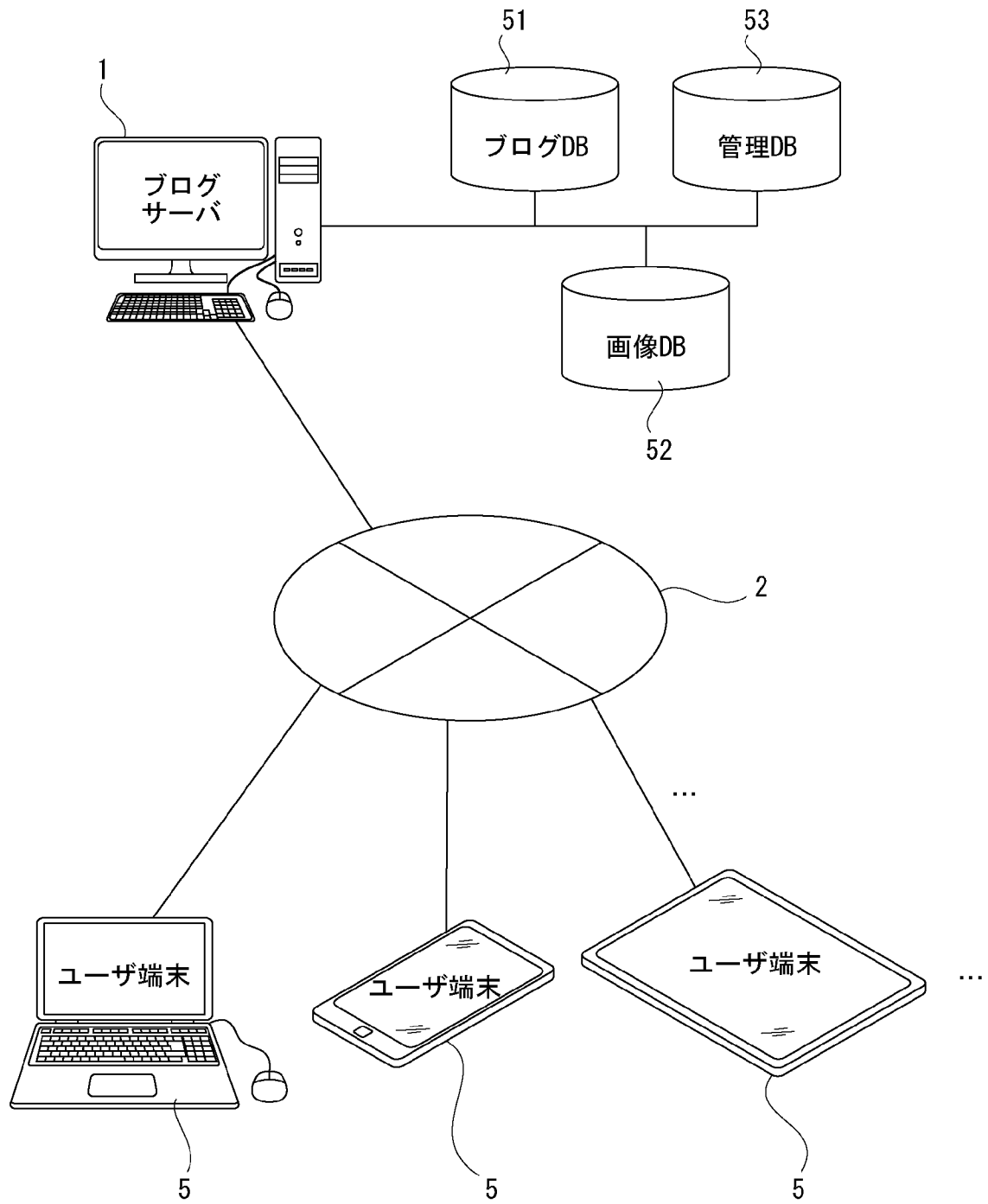
1 または複数の記事が含まれるブログの人気度指数を取得する人気度指数取得機能と、

前記ブログごとの前記人気度指数に応じて該ブログごとに圧縮するか否かを判定するための閾値を設定する閾値設定機能と、

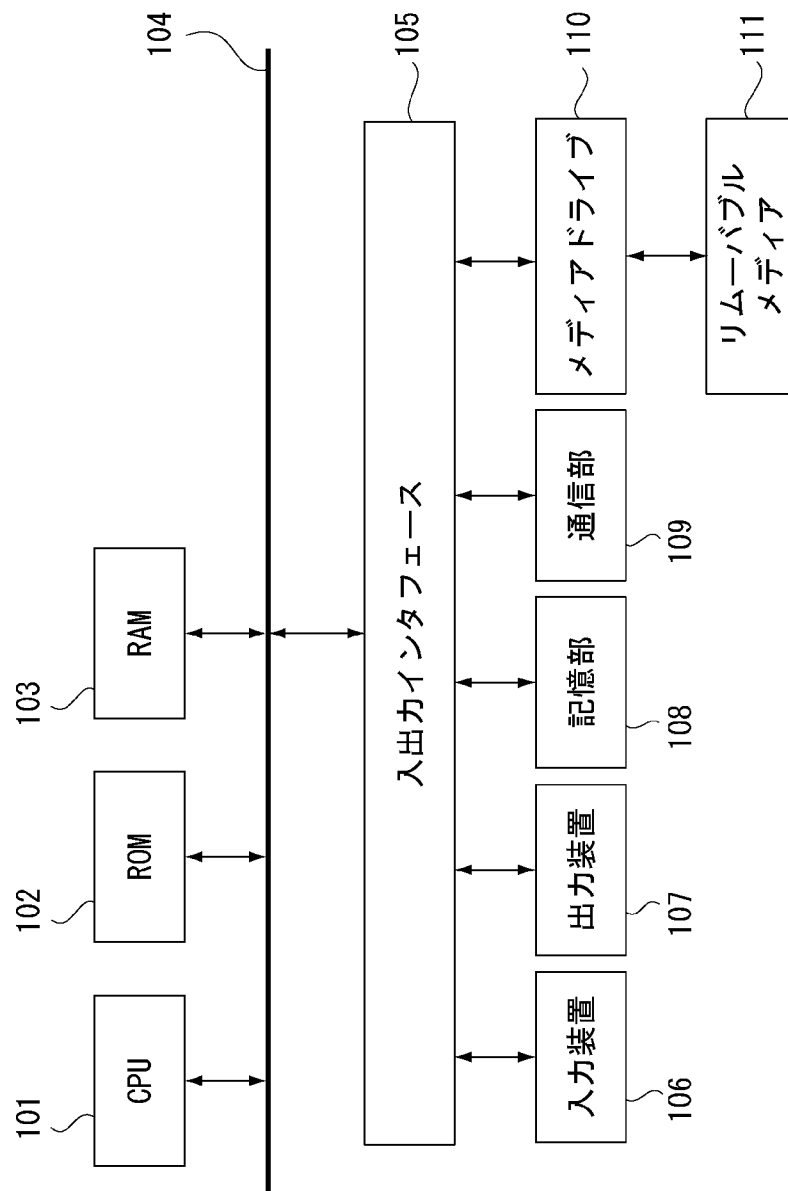
一つの前記ブログに含まれる記事の総データ量と前記閾値とに基づいて当該一つの前記ブログに含まれる記事の少なくとも一部を圧縮する圧縮対象ブログとするか否かを判定し、前記圧縮対象ブログに含まれる各記事の中から圧縮対象とする記事を圧縮対象記事として判定する判定機能と、を

情報処理装置に実行させるプログラムを記憶した記憶媒体。

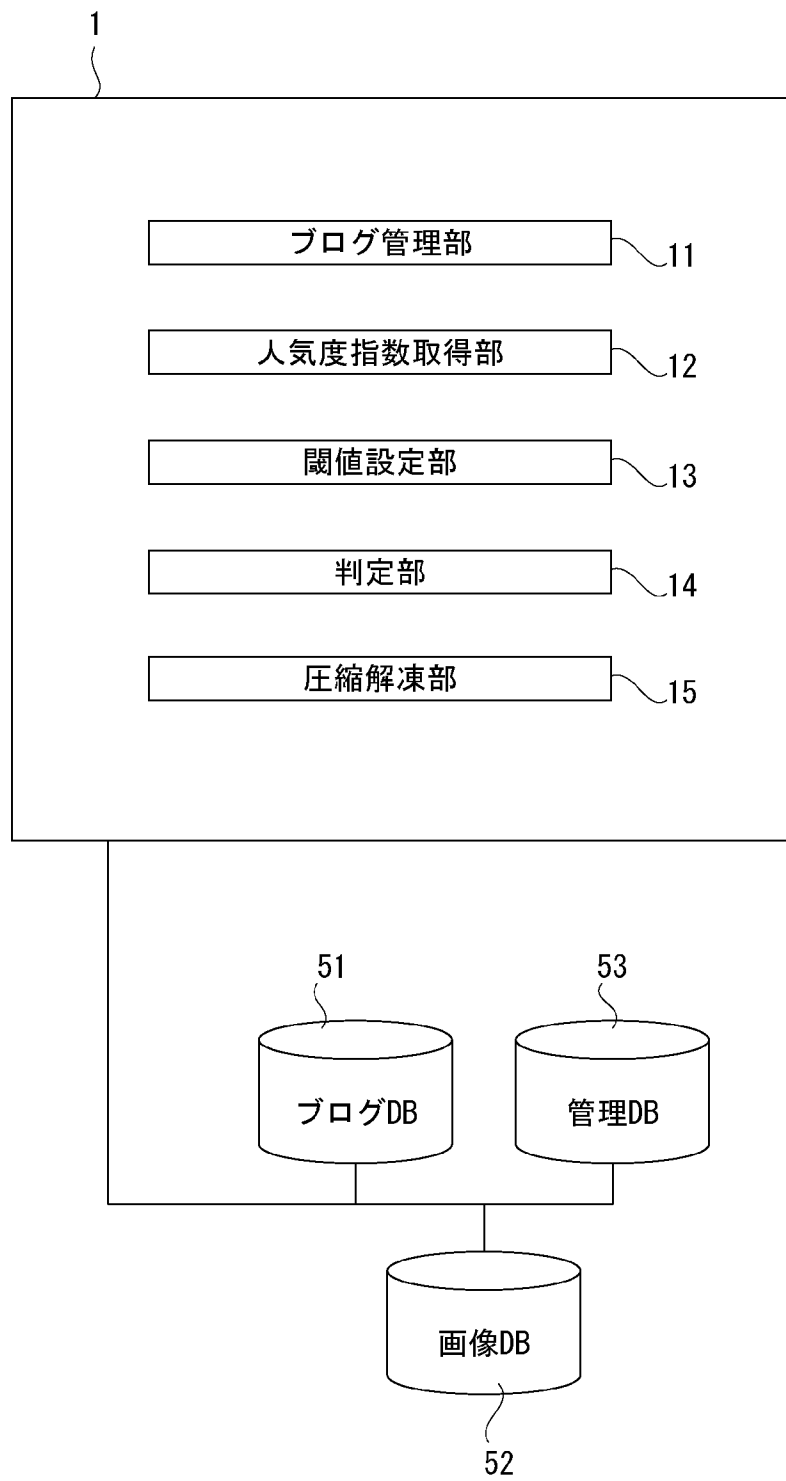
[図1]



[図2]



[図3]

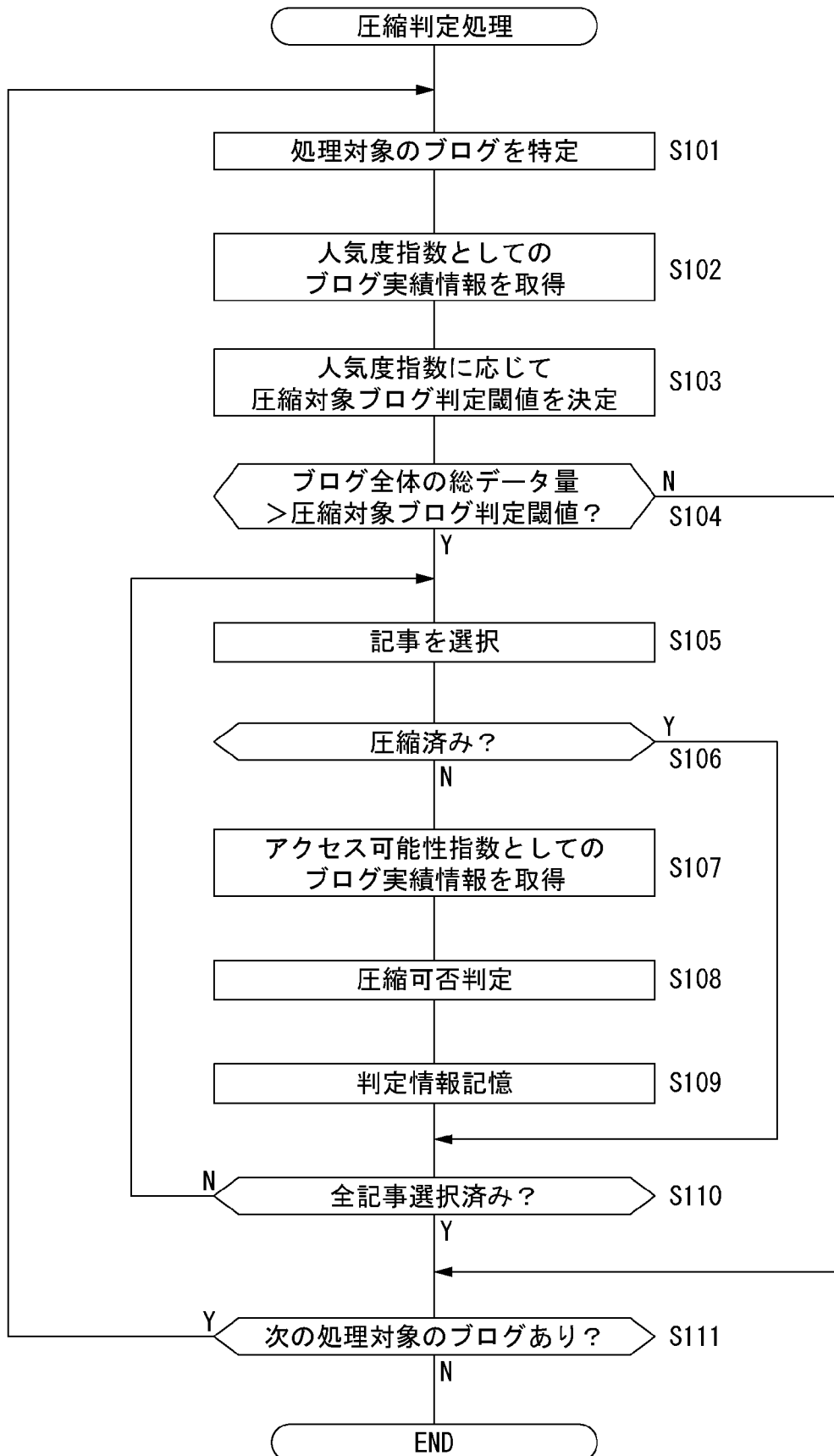


[図4]

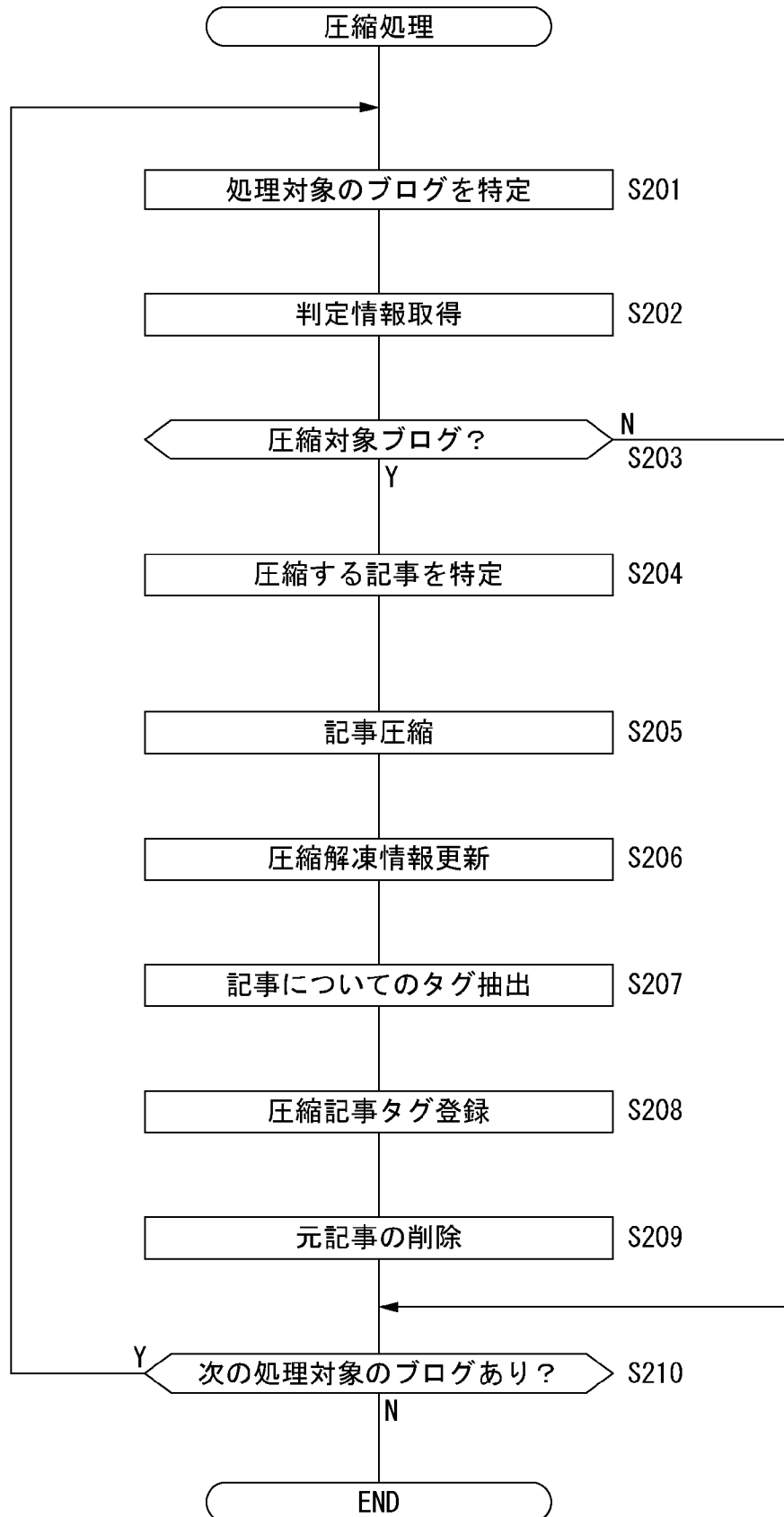
管理DB 53

ブログID	ユーザ情報	ブログ管理情報	ブログ実績情報	サイズ情報	判定情報	圧縮解凍情報	圧縮記事タグ
::	::	::	::	::	::	::	::

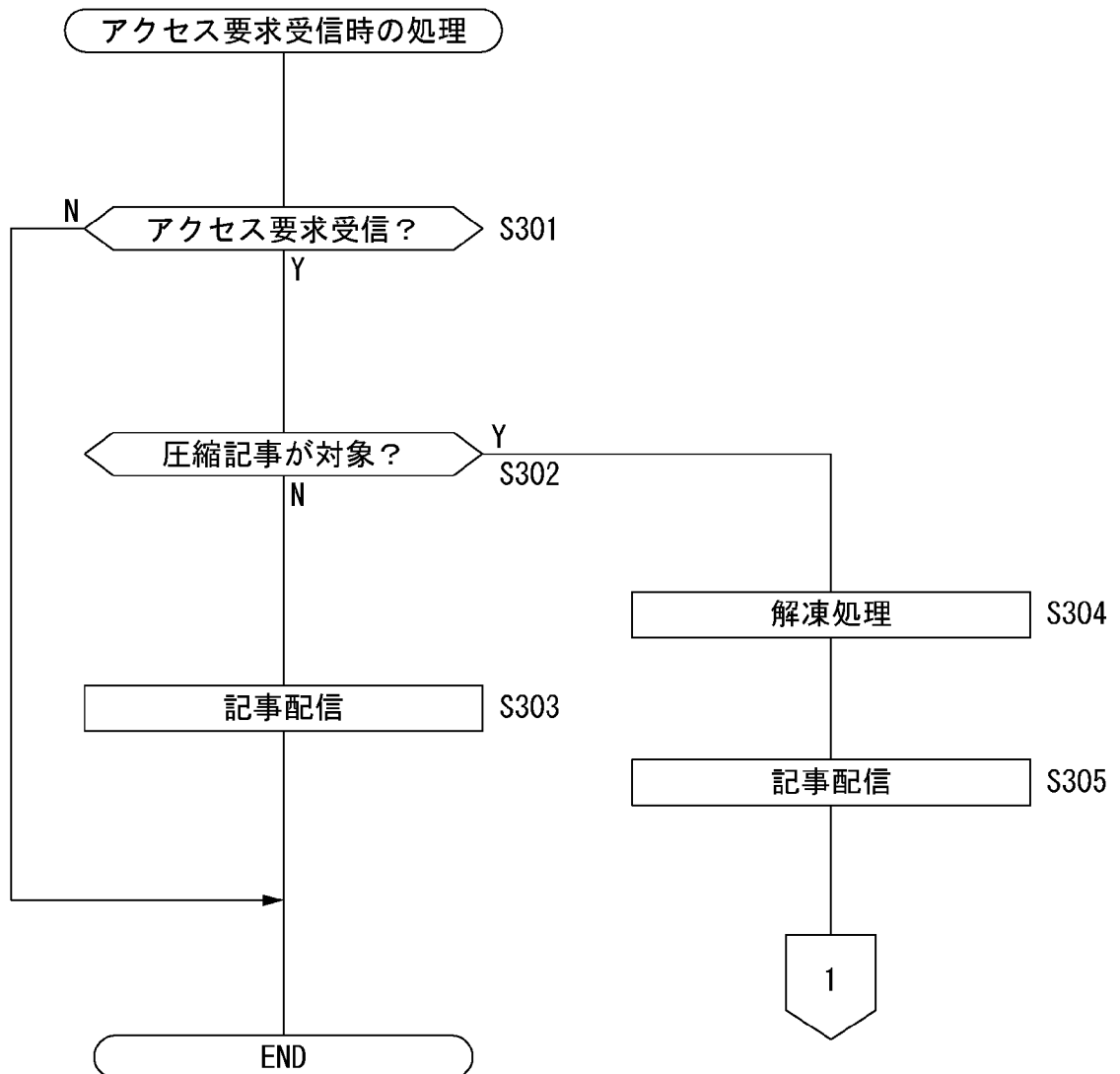
[図5]



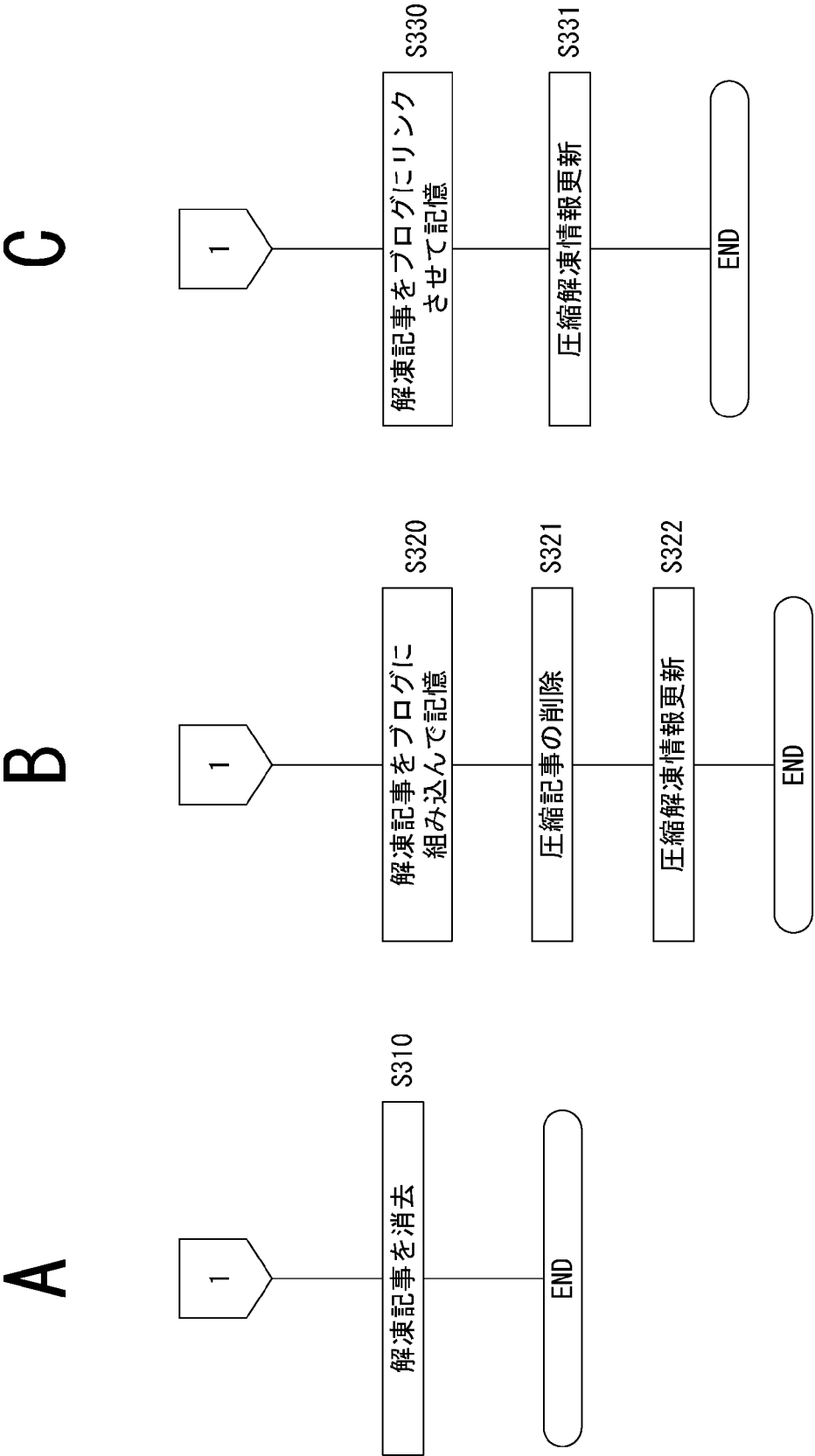
[図6]



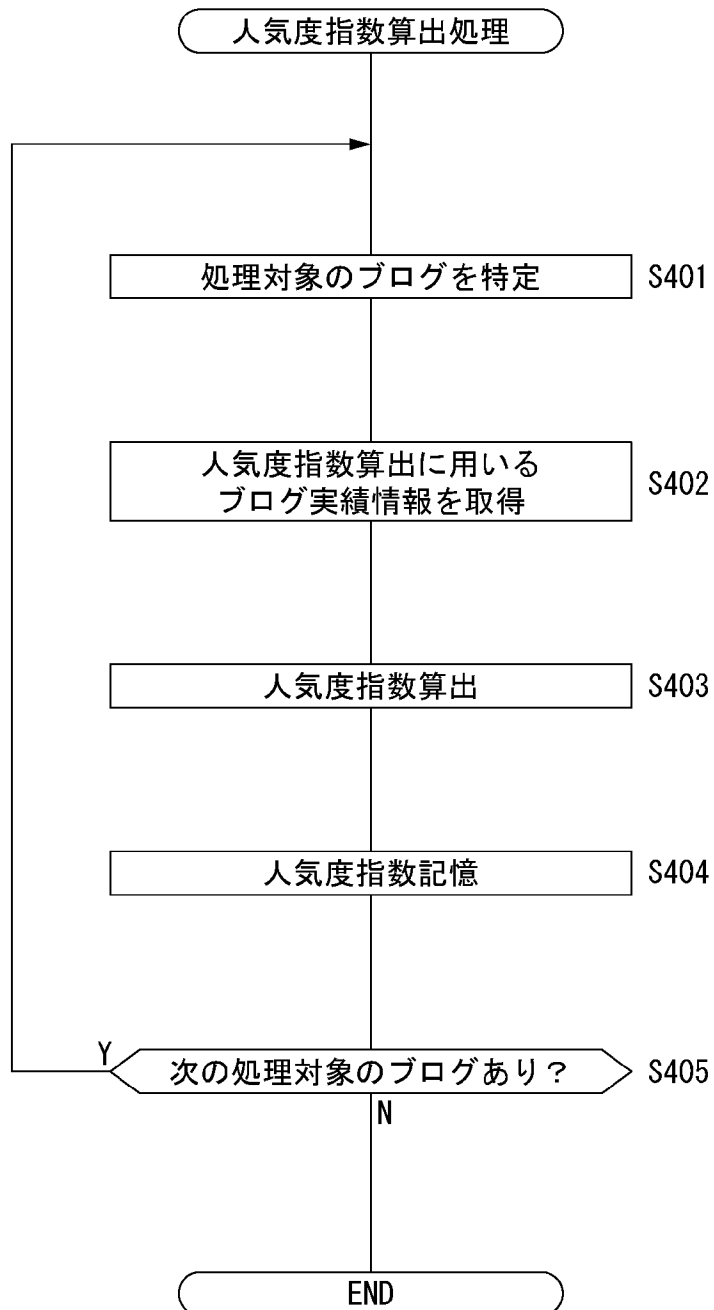
[図7]



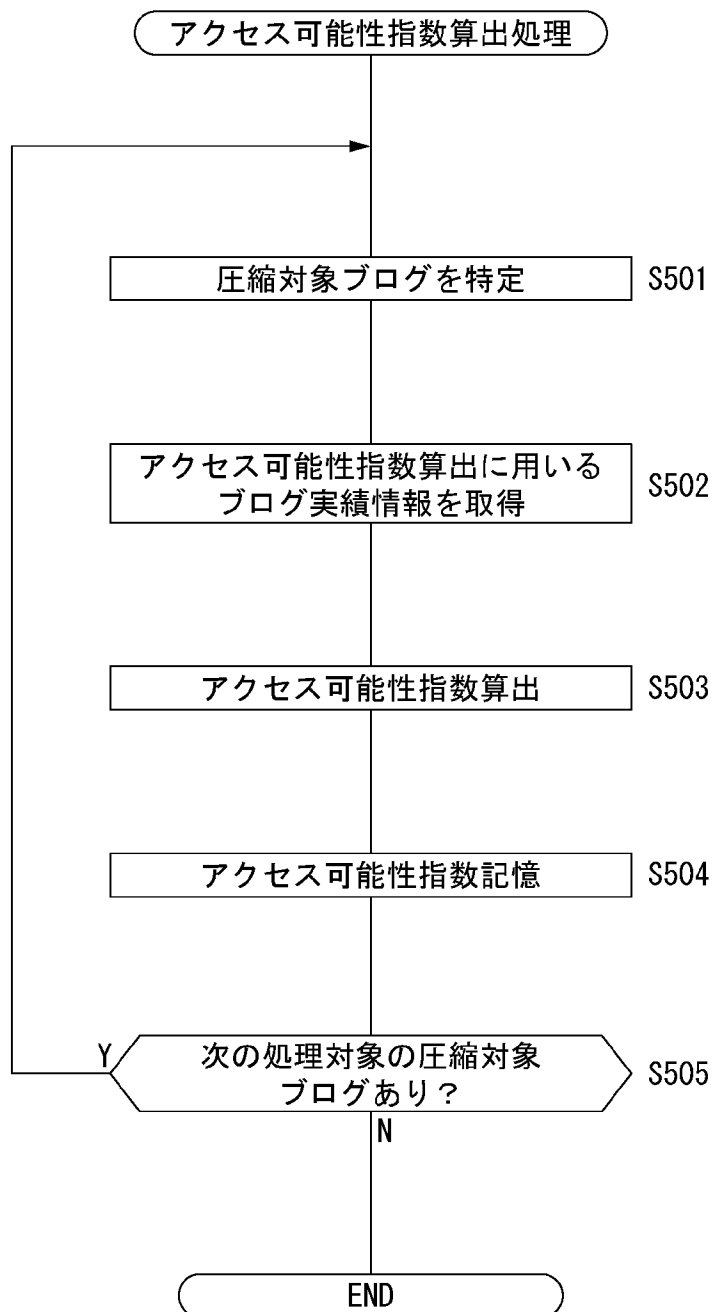
[図8]



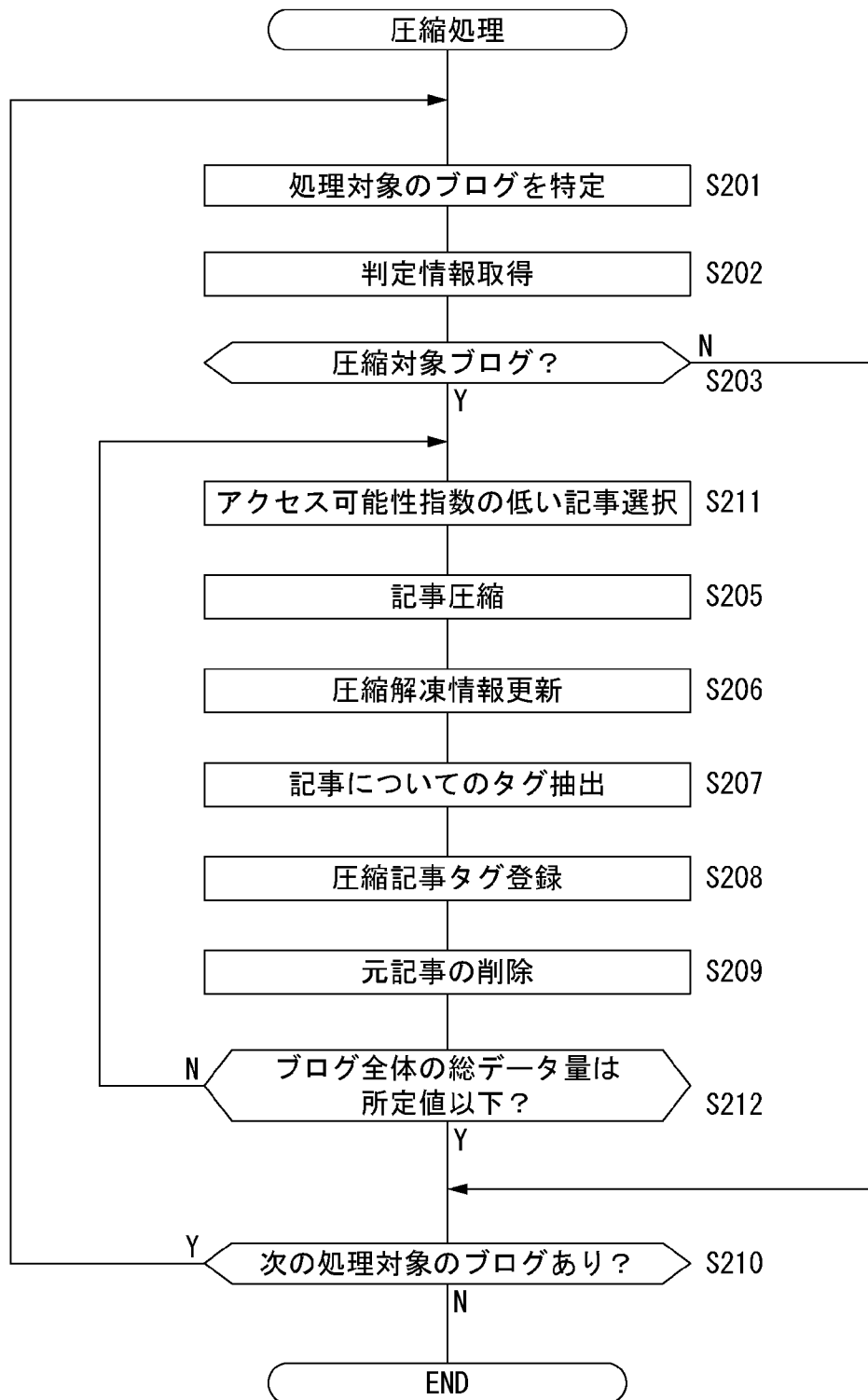
[図9]



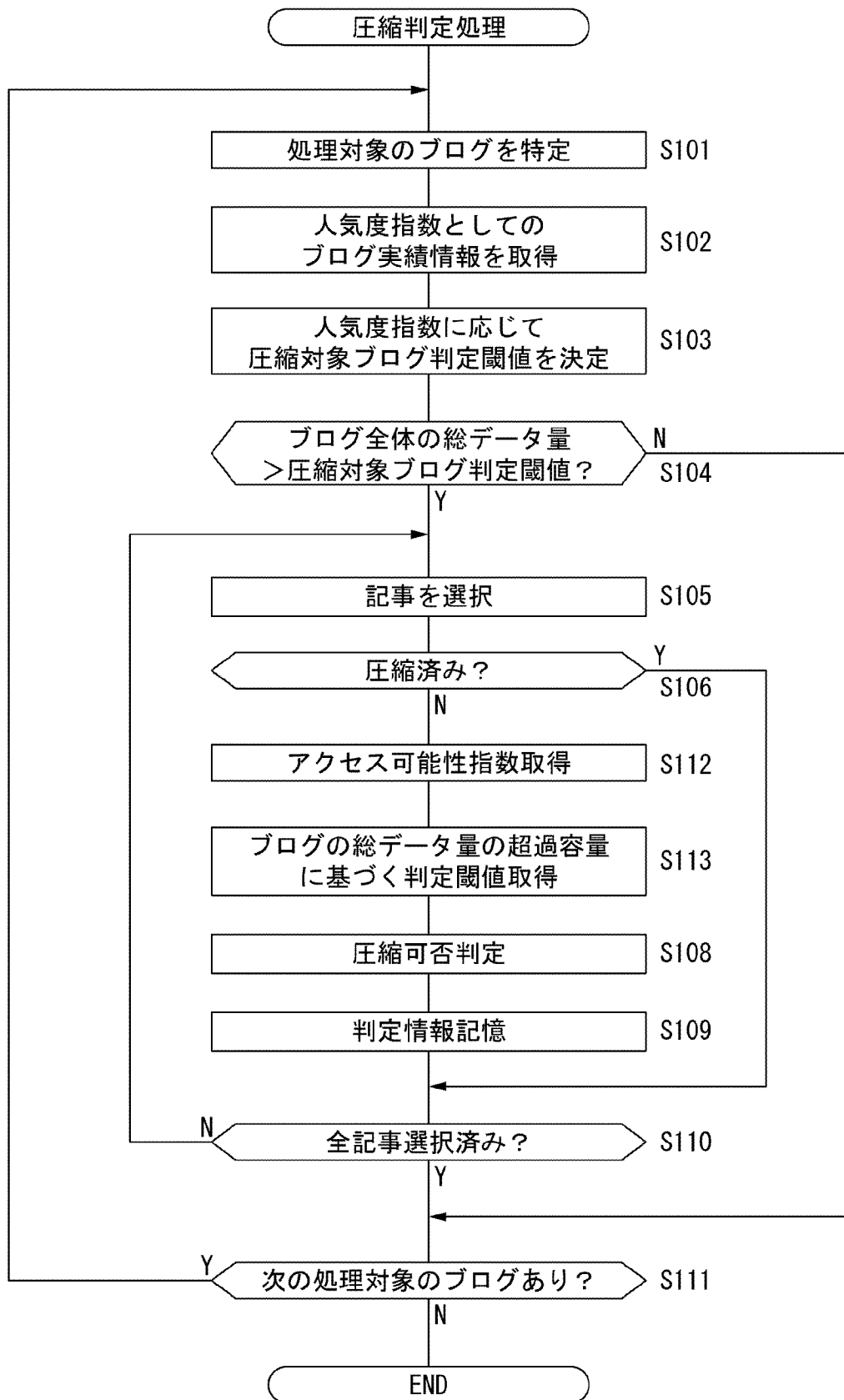
[図10]



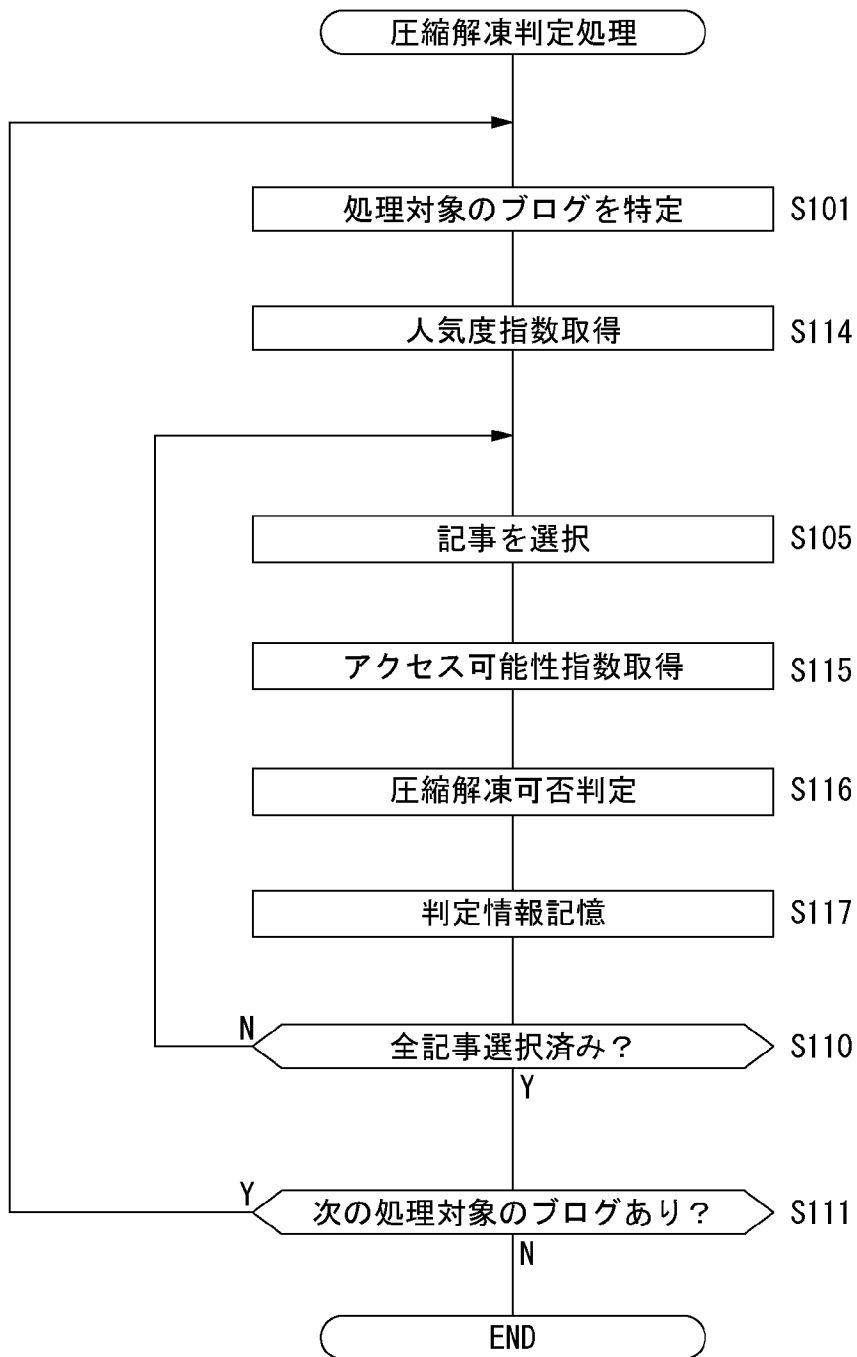
[図11]



[図12]



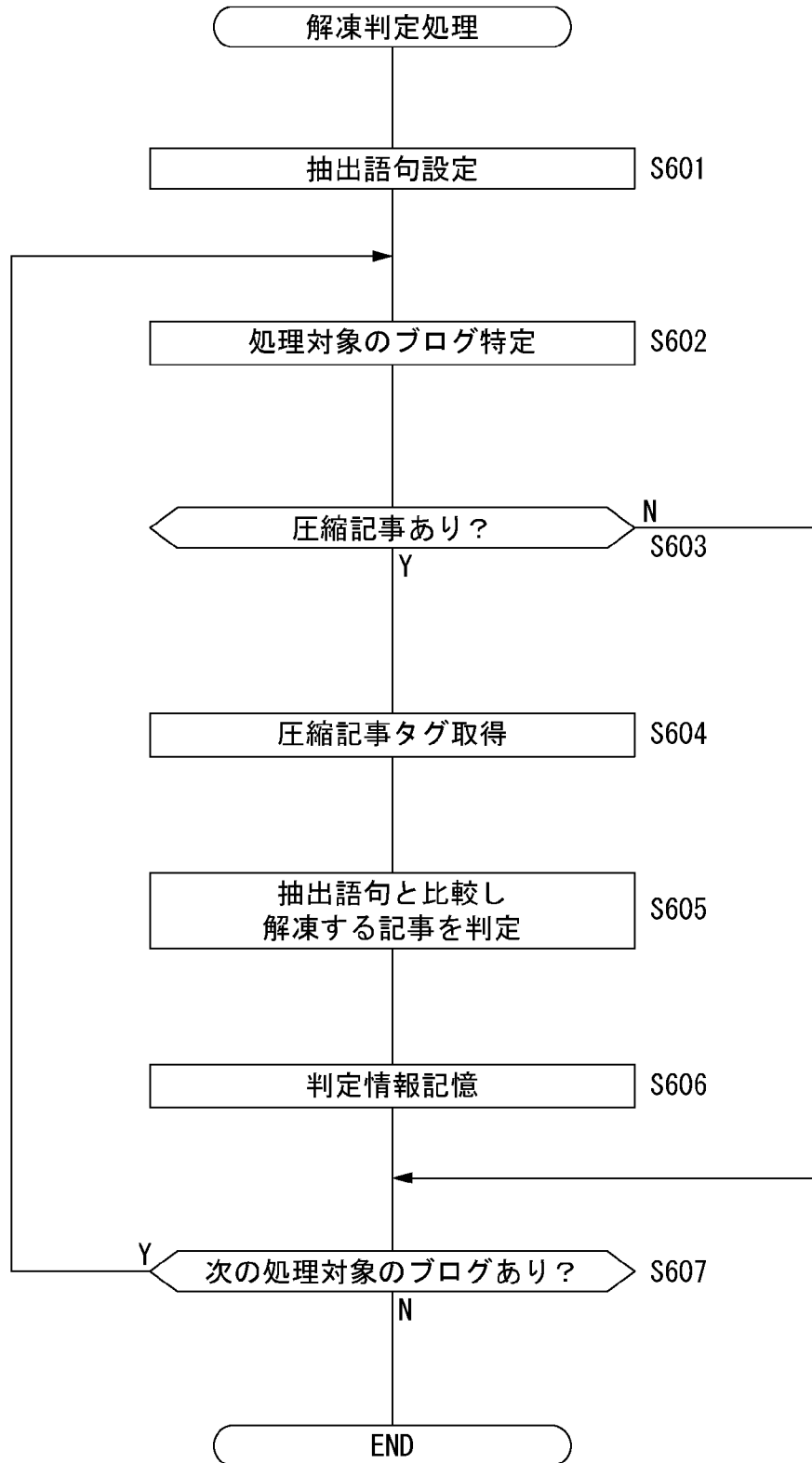
[図13]



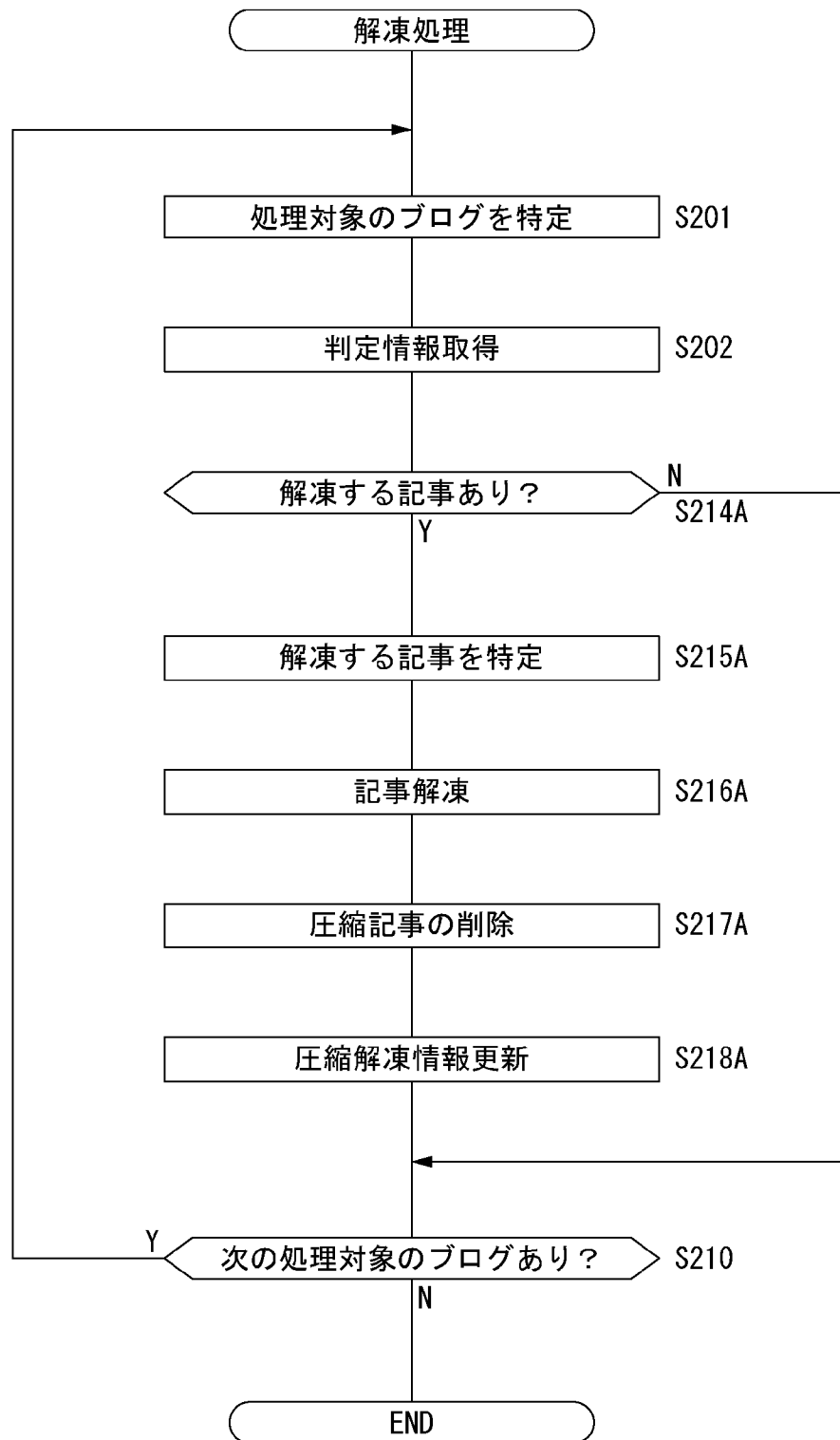
[図14]



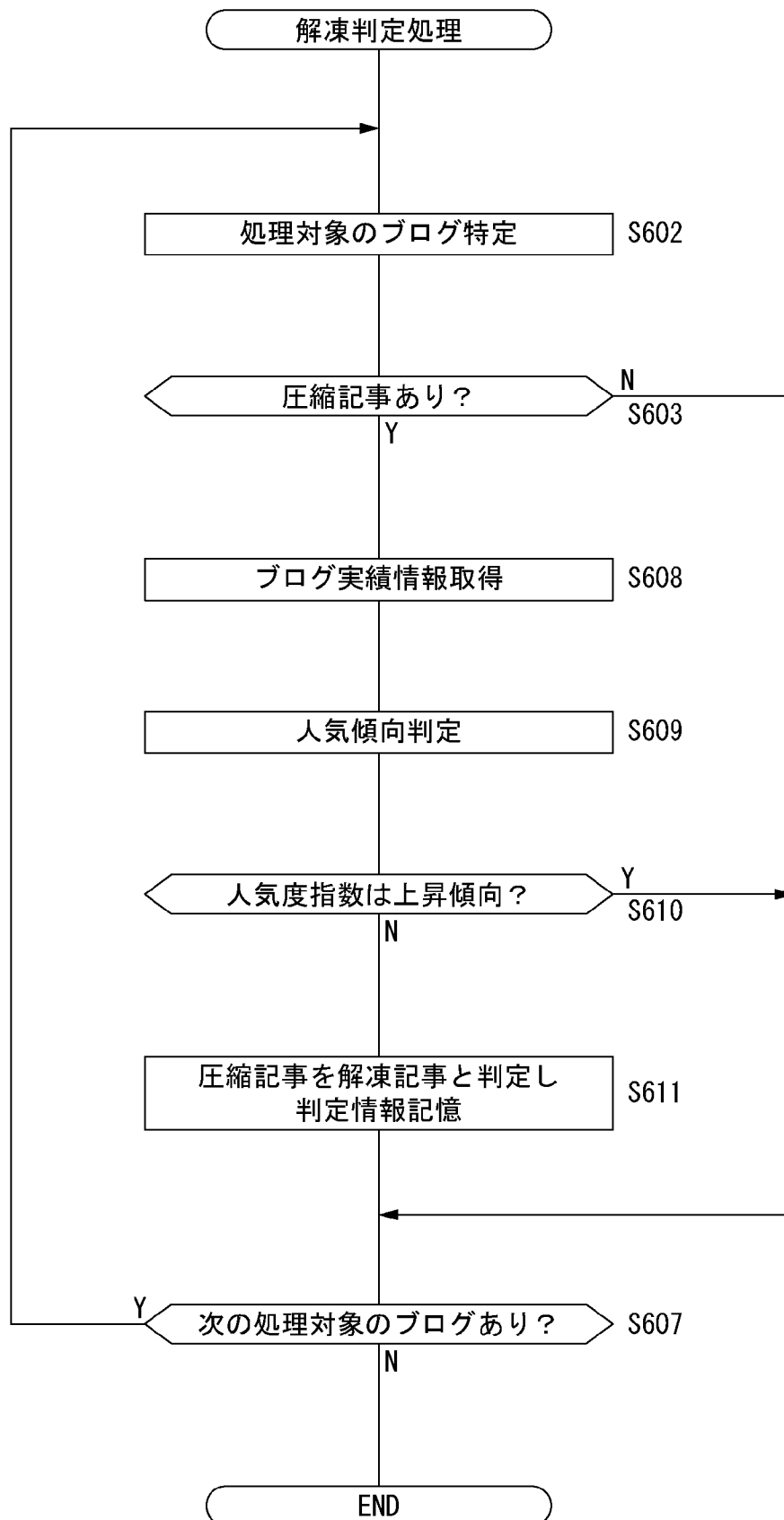
[図15]



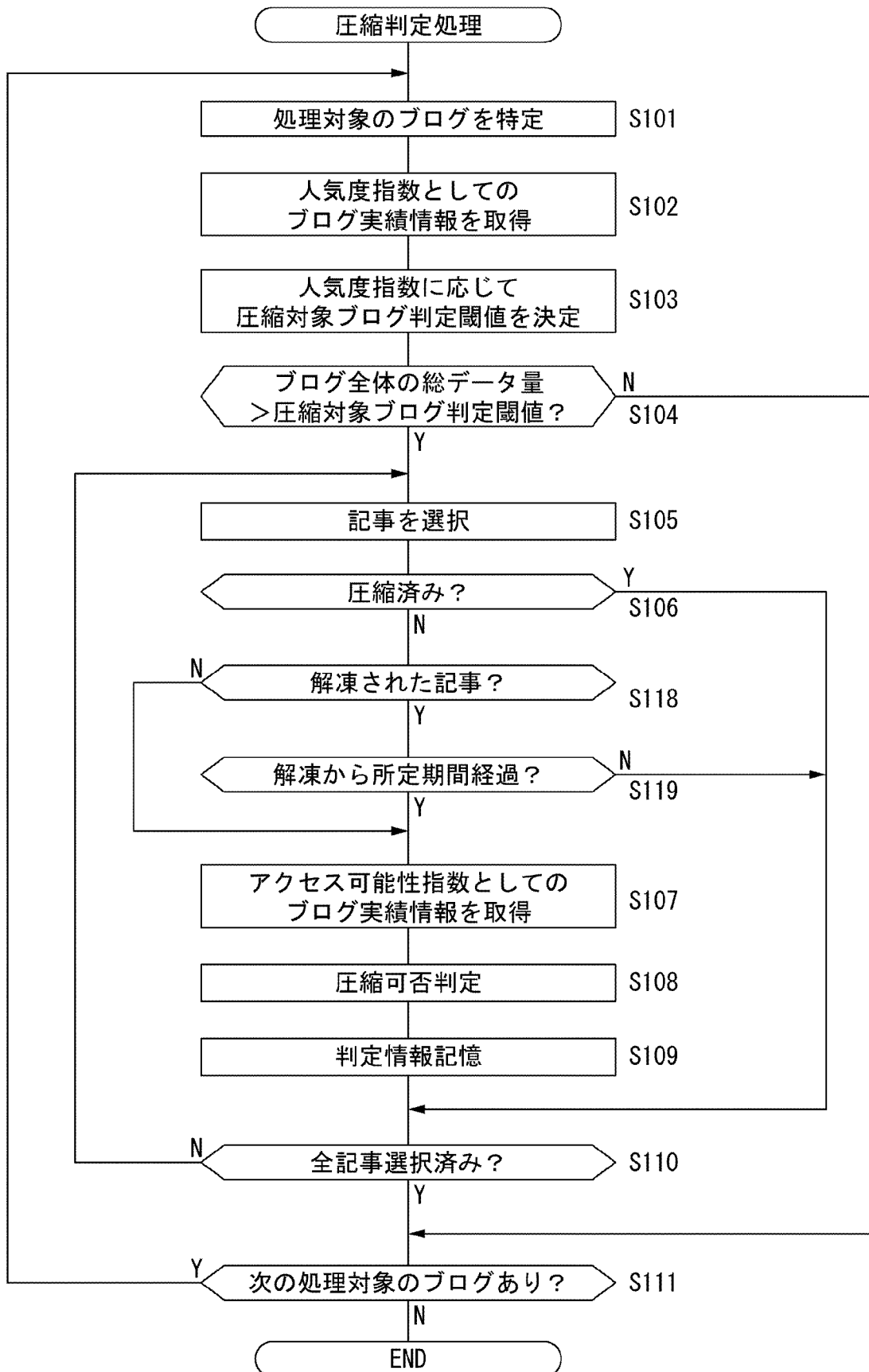
[図16]



[図17]



[図18]



INTERNATIONAL SEARCH REPORT

International application No.

PCT/JP2016/081142

A. CLASSIFICATION OF SUBJECT MATTER

G06F12/00(2006.01) i

According to International Patent Classification (IPC) or to both national classification and IPC

B. FIELDS SEARCHED

Minimum documentation searched (classification system followed by classification symbols)

G06F12/00

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

Jitsuyo Shinan Koho	1922-1996	Jitsuyo Shinan Toroku Koho	1996-2016
Kokai Jitsuyo Shinan Koho	1971-2016	Toroku Jitsuyo Shinan Koho	1994-2016

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

C. DOCUMENTS CONSIDERED TO BE RELEVANT

Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
A	JP 2009-70361 A (Hitachi, Ltd.), 02 April 2009 (02.04.2009), & US 2009/0070356 A1	1-10
A	US 2011/0138270 A1 (INTERNATIONAL BUSINESS MACHINES CORP.), 09 June 2011 (09.06.2011), & CN 102053990 A	1-10
A	JP 2012-222504 A (Sony Corp.), 12 November 2012 (12.11.2012), & US 2012/0257872 A1 & CN 102740033 A	1-10
A	JP 2012-14236 A (NTT Docomo Inc.), 19 January 2012 (19.01.2012), (Family: none)	1-10

☐ 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
22 December 2016 (22.12.16)

Date of mailing of the international search report
10 January 2017 (10.01.17)

Name and mailing address of the ISA/
Japan Patent Office
3-4-3, Kasumigaseki, Chiyoda-ku,
Tokyo 100-8915, Japan

Authorized officer

Telephone No.

A. 発明の属する分野の分類 (国際特許分類 (IPC))

Int.Cl. G06F12/00(2006.01)i

B. 調査を行った分野

調査を行った最小限資料 (国際特許分類 (IPC))

Int.Cl. G06F12/00

最小限資料以外の資料で調査を行った分野に含まれるもの

日本国実用新案公報	1922-1996年
日本国公開実用新案公報	1971-2016年
日本国実用新案登録公報	1996-2016年
日本国登録実用新案公報	1994-2016年

国際調査で使用した電子データベース (データベースの名称、調査に使用した用語)

C. 関連すると認められる文献

引用文献の カテゴリー*	引用文献名 及び一部の箇所が関連するときは、その関連する箇所の表示	関連する 請求項の番号
A	JP 2009-70361 A (株式会社日立製作所) 2009.04.02 & US 2009/0070356 A1	1-10
A	US 2011/0138270 A1 (INTERNATIONAL BUSINESS MACHINES CORPORATION) 2011.06.09 & CN 102053990 A	1-10
A	JP 2012-222504 A (ソニー株式会社) 2012.11.12 & US 2012/0257872 A1 & CN 102740033 A	1-10

C欄の続きにも文献が列挙されている。

パテントファミリーに関する別紙を参照。

* 引用文献のカテゴリー

「A」特に関連のある文献ではなく、一般的技術水準を示すもの

「E」国際出願日前の出願または特許であるが、国際出願日以後に公表されたもの

「L」優先権主張に疑義を提起する文献又は他の文献の発行日若しくは他の特別な理由を確立するために引用する文献 (理由を付す)

「O」口頭による開示、使用、展示等に言及する文献

「P」国際出願日前で、かつ優先権の主張の基礎となる出願

の日の後に公表された文献

「T」国際出願日又は優先日後に公表された文献であって出願と矛盾するものではなく、発明の原理又は理論の理解のために引用するもの

「X」特に関連のある文献であって、当該文献のみで発明の新規性又は進歩性がないと考えられるもの

「Y」特に関連のある文献であって、当該文献と他の1以上の文献との、当業者にとって自明である組合せによって進歩性がないと考えられるもの

「&」同一パテントファミリー文献

国際調査を完了した日

22.12.2016

国際調査報告の発送日

10.01.2017

国際調査機関の名称及びあて先

日本国特許庁 (ISA/J P)

郵便番号 100-8915

東京都千代田区霞が関三丁目4番3号

特許庁審査官 (権限のある職員)

大塚 俊範

5 B

5583

電話番号 03-3581-1101 内線 3545

C (続き) . 関連すると認められる文献		
引用文献の カテゴリー*	引用文献名 及び一部の箇所が関連するときは、その関連する箇所の表示	関連する 請求項の番号
A	JP 2012-14236 A (株式会社エヌ・ティ・ティ・ドコモ) 2012.01.19 (ファミリーなし)	1-10