

(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 2018/073941 A1

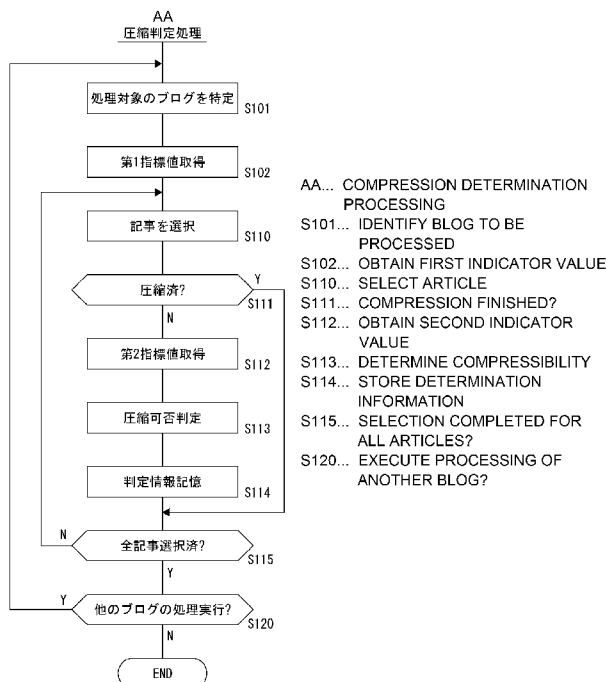
- (51) 国際特許分類:
G06F 12/00 (2006.01)
- (21) 国際出願番号: PCT/JP2016/081143
- (22) 国際出願日: 2016年10月20日(20.10.2016)
- (25) 国際出願の言語: 日本語
- (26) 国際公開の言語: 日本語
- (71) 出願人: 楽天株式会社 (RAKUTEN, INC.) [JP/JP]; 〒1580094 東京都世田谷区玉川一丁目14番1号 Tokyo (JP).
- (72) 発明者: 山本 陽司 (YAMAMOTO, Yoji); 〒1580094 東京都世田谷区玉川一丁目14番1号 楽天株式会社内 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, 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,

(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

[図5]



(57) Abstract: An increase in the size and/or number of blogs in a server for providing a blog service leads to the gradual exhaustion of memory resources. In response, the present invention reduces this memory load. To this end, the present invention obtains a first indicator value expressing the popularity of the blog overall, and a second indicator value expressing the accessibility of each article contained in the blog. On the basis of the first and second indicator values, the present invention determines for each article whether or not to compress the article.

(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。そこでブログ全体の人気度を示す第1指標値と、そのブログに含まれる記事ごとのアクセス可能性を示す第2指標値を取得する。そして第1指標値と第2指標値に基づいて、記事ごとに、その記事を圧縮するか否かを判定する。

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)）

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1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り、非公開のものもある。

多くのユーザは、自己発信のツールとしたり、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も多い。一方で頻繁に記事を更新して人気を集めるユーザまで玉石混交であり、ブログや個々の記事におけるアクセス数はさまざま

ある。

例えば、あるユーザがブログを開設して2、3の記事をアップしたものの、その後更新せず、或いはログインもせず、アクセスもないまま長期間放置されているようなブログも存在する。このようなブログはユーザにとっても不要である場合があり、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っている。

[0005] そこで、アクセス可能性の低い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。

ただし圧縮や解凍の処理は負荷が高い。また、アクセスの際に解凍が必要となると、その処理時間により閲覧ユーザにページアクセスの遅れ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、できるだけ圧縮や解凍の処理回数を減らしたい。

そこで本発明は、圧縮する記事を適切に選択し、なるべく圧縮や解凍の処理を少なくしつつ、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又は複数の記事が含まれるブログの人気度を示す第1指標値を取得する第1取得部と、前記ブログに含まれる複数の記事ごとに、当該記事に対するアクセス可能性を示す第2指標値を取得する第2取得部と、ブログに含まれる記事について、前記第1指標値と前記第2指標値とに基づいて、圧縮するか否かを判定する判定部と、を備える。

ブログに含まれる記事の中で、アクセス可能性の小さい記事（例えば不人気記事）を圧縮することを考える。このときに、アクセス可能性の小さい記事が、人気の高いブログ内の記事であるか、人気の低いブログ内の記事であるかも考慮して、圧縮するか否かを判定する。

[0007] 上記した情報処理装置においては、前記判定部によって圧縮すると判定された記事について圧縮を行い、また既に圧縮された記事に対してのアクセス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ることが考えられる。

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においては、前記判定部は、前記第 1 指標値と前記第 2 指標値とに基づいて、既に圧縮した記事を解凍するか否かを判定することが考えられる。

一旦圧縮した記事についても、その後、定期的又は不定期に第 1、第 2 指標値を取得して、アクセス可能性の変化を確認する。もし最新の状況としてアクセス可能性が高くなっていたら、解凍しておくように判定する。

[0009] 上記した情報処理装置においては、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ことが考えられる。

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。

[0010] 上記した情報処理装置においては、前記判定部は、ブログの人気度の変化を監視し、或るブログについて人気度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の全部又は一部を解凍する記事と判定することが考えられる。

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においては、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、前記判定部は、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ことが考えられる。

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。

[0012] 上記した情報処理装置においては、前記第 1 指標値は、ブログ全体につい

て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、ブログに設定された総被リンク数、ブログに記述された総コメント数、ブログのページランクを示す値、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、の少なくとも一部に基づいて求められた値であることが考えられる。

これらの値はブログ全体の人気度に応じた値となりやすい。

[0013] 上記した情報処理装置においては、前記第2指標値は、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、記事に設定された被リンク数、記事のページランクを示す値、記事のページビューの増加傾向を示す値、記事のアクセスのない期間の長さを示す値、特定語句の有無を示す値、の少なくとも一部に基づいて求められた値であることが考えられる。

これらの値は記事毎のアクセス可能性に応じた値となりやすい。

[0014] 本発明に係る情報処理方法は、1又は複数の記事が含まれるブログの人気度を示す第1指標値を取得するステップと、前記ブログに含まれる複数の記事ごとに、当該記事に対するアクセス可能性を示す第2指標値を取得するステップと、ブログに含まれる記事について、前記第1指標値と前記第2指標値とに基づいて、圧縮するか否かを判定するステップと、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

[0015] 本発明によれば、記事の圧縮によって記憶リソースの圧迫を抑えけるとともに、なるべく圧縮や解凍の処理機会が少なくなるように、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。

図面の簡単な説明

[0016] [図1]本発明の実施の形態の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の実施の形態の圧縮判定のための処理の変形例のフローチャートである。

[図10]第3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11]第3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12]第4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13]第4の実施の形態の解凍処理のフローチャートである。

[図14]第5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15]第6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

[0017] 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. 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. まとめ及び変形例 >

< 10. プログラム及び記憶媒体 >

[0018] 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（又は限定した範囲）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。

[0019] “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1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1つの話題としての記事ではなく、1又は複数の話題について1つのURLで閲覧される記事群を指すものと考えても良い。

[0020] “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、他人又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。

[0021] “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。

[0022] < 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。

[0023] 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 の全部又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。

[0024] 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。既にブログ開設済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。

[0025] 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。

[0026] 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。

[0027] 図2において、コンピュータ装置のCPU (Central Processing Unit) 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。RAM 103にはまた、CPU 101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。

CPU 101、ROM 102、およびRAM 103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、入力部106、出力部107、記憶部

108、通信部109が接続されている。

入力部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。

出力部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。

[0028] 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。

[0029] <2. ブログサーバ及びデータベース>

図3に1又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又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。１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、１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。さらに機能の１つが、複数の情報処理装置によって実現されてもよい。

[0030] ブログサーバ１は、図示するようにブログ管理部１１、第１取得部１２、第２取得部１３、判定部１４、圧縮解凍部１５としての機能を備える。

[0031] 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。

[0032] 第１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第１指標値を取得する処理を行う。

ブログについての第１指標値とは、ブログ全体について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、ブログに設定された総被リンク数、ブログに記述された総コメント数、ブログのページランク、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、ブログに掲載された広告に対するクリック数などである。

[0033] 第２取得部１３は、同じく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、ブログに含まれる複数の記事ごとに、当該記事に対するアクセス可能性を示す第2指標値を取得する処理を行う。

第2指標値としては、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、記事に設定された被リンク数、記事のページランクを示す値、記事のページビューの増加傾向を示す値、記事のアクセスのない期間の長さを示す値、記事とともに掲載された広告に対するクリック数、特定語句の有無を示す値などである。

[0034] 判定部14は、ブログに含まれる記事について、第1指標値と第2指標値とに基づいて、圧縮するか否かを判定する処理を行う。

また判定部14は、圧縮された記事について、第1指標値と第2指標値とに基づいて、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。

[0035] 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が発生した場合に、圧縮に対する解凍処理を行う。

さらに圧縮解凍部15は、既に圧縮された記事について判定部14によって解凍すると判定された記事を解凍する処理を行う。

[0036] 図3ではブログサーバ1がアクセスするDBとしてブログDB51、画像DB52、管理DB53を示している。

ブログDB51は、各記述者についてのブログデータを保存するDBである。各ブログのウェブページデータが保存され、また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は、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。

[0037] 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。

[0038] 管理DB53は、各ブログの管理の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。

1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。例えば各ブログ（ブログID）毎に、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。

[0039] 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。

[0040] 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。

[0041] 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。つまり上述の第1指標値や第2指標値である。

従ってブログの人気の指標となる情報（第1指標値）として、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、総コメント数、ブログのページランク、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、広告クリック数などが逐次更新されていく。

これらの値はブログ全体の人気度に応じた値となり第1指標値として適している。

[0042] またブログ実績情報として、各記事のアクセス可能性の指標となる情報（第2指標値）が各記事に対応して記憶される。例えば各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、被リンク数、記事のページランクを示す値、記事のページビューの増加傾向を示す値、記事のアクセスのない期間の長さを示す値、特定語句の有無を示す値などである。

これらの値は記事毎のアクセス可能性に応じた値となり、第2指標値として適している。

[0043] サイズ情報は、そのブログ全体のデータサイズの情報である。また各記事のサイズ情報を記憶しても良い。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に記憶したデータサイズと画像DB52に記憶した画像データのデータサイズを合わせて管理してもよい、それぞれ別個に管理してもよい。

[0044] 判定情報は、各記事について判定部14が第1指標値、第2指標値に基づいて判定した、圧縮するか否かの情報である。また圧縮した記事について判定部14が判定した、解凍するか否かの情報も含む。即ち圧縮可／不可や解凍可／不可を示す情報である。これらは例えばフラグデータとして更新される。

なお、圧縮可／不可や解凍可／不可を示すフラグは、その後の圧縮処理や解凍処理で圧縮や解凍とすべきことを示す情報となる。従ってこれらのフラグは圧縮や解凍が行われた際にはクリアされればよい。

また判定情報には、ブログの人気度の判定結果の情報を含む場合もある。

[0045] 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。

[0046] 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。

[0047] 以上の各DB（ブログDB51、画像DB52、管理DB53）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、各DBの一部又は全部が別体、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1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1つのDBの形態で例示したものに過ぎない。

[0048] <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では、ブログサーバ1は、アクセス可能性の低い記事については圧縮して保存する。圧縮した記事にアクセスがあった場合には、解凍して配信する。

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が発生しないようにしたい。

そこで本実施の形態では、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とする。

[0049] 図5はブログサーバ1が実行する圧縮判定処理例を示している。

なお、図5以降、後述する図15までのフローチャートで示す処理は、ブログサーバ1が図3に示したブログ管理部11、第1取得部12、第2取得部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。

[0050] 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又は一部のブログに対して実行する。これは、記事毎に圧縮可否を判定する処理である。

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1つのブログを特定する。例えば順番に1つのブログを選択するものとすればよい。

[0051] ステップS102でブログサーバ1は、処理対象としたブログについての第1指標値を取得する。例えば管理DB53から、当該ブログ（ブログID）について記憶されているブログ実績情報に含まれる第1指標値を読み出す。

具体的には、当該ブログの、総ページビュー数、アクセスしたユニークユーザー数、総被リンク数、総コメント数、ブログのページランク、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などの全部又は一部を読み出す。そしてこれらを当該ブログの人気度を測る材料とする。

[0052] そしてブログサーバ1は、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S110～S115の処理を行う。

まずステップS110でブログサーバ1は、ブログ内の1つの記事を選択する。

ステップS111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、圧縮可否判定は不要なためステップS115に進む。ステップS115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S110に戻って次の記事を選択する。

なお、ステップS115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。

[0053] ステップS 1 1 0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 1 1からS 1 1 2に進み、当該記事についての第2指標値を取得する。例えば管理DB 5 3から、当該ブログについて記憶されているブログ実績情報に含まれる、当該記事の情報を読み出す。例えば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、被リンク数、当該記事のページランクを示す値、当該記事のページビューの増加傾向を示す値、当該記事のアクセスのない期間の長さを示す値、当該記事とともに掲載された広告のクリック数などである。

[0054] そしてブログサーバ1はステップS 1 1 3で、当該記事について圧縮可否判定を行う。ここではアクセス可能性が大きいか、小さいかを判定する。そして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。

ブログサーバ1はこの判定をステップS 1 0 2で取得した第1指標値と、ステップS 1 1 2で取得した第2指標値に基づいて行う。

具体的には、記事自体のアクセス可能性のみで圧縮可否を判断せず、第1指標値によりブログの人気度を判定し、それを反映させた圧縮可否判定を行う。

[0055] 例えばまず第1指標値により、ブログ自体の人気度を判定する。例えばページビュー数などに基づいて、人気度を複数段階のいずれかに分類する。例えば「人気／不人気」の2段階でもよいし、「人気／普通／不人気」の3段階でも良い。また人気レベルとして4段階以上に分類しても良い。

このような人気度の判定結果を反映させて、記事のアクセス可能性の指標を確認し、圧縮可否を判定する。

[0056] 例えば、第1指標値を総ページビュー数（PV）とし、

- ・ PVが0－1 0 0 0 0・・・不人気ブログ
- ・ PVが1 0 0 0 1－1 0 0 0 0 0 0・・・普通ブログ
- ・ PVが1 0 0 0 0 0 1以上・・・人気ブログ

とする。

第2指標値をアクセスのない期間の長さを示す値とする。

- ・不人気ブログの場合・・・アクセスのない期間が1年以上であれば圧縮
 - ・普通ブログの場合・・・アクセスのない期間が3年以上であれば圧縮
 - ・人気ブログの場合・・・アクセスのない期間が5年以上であれば圧縮
- とする。

[0057] このように、まずブログの人気度を判定し、その人気度によって、アクセス可能性による圧縮可の判定基準を変更する。

これによってブログ自体の人気度を考慮したうえで、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S113でブログサーバ1は、ブログの人気度を反映させた状態での記事のアクセス可能性の指標から、圧縮可否判定を行うようにしている。

[0058] なおアクセス可能性を示す第2指標値として他の指標を用いても良い。例えば第2指標値の記事毎のユニークユーザ数UNとする。そして、

・不人気ブログの場合・・・ $UN < UN1$ であれば圧縮

・普通ブログの場合・・・ $UN < UN2$ であれば圧縮

・人気ブログの場合・・・ $UN < UN3$ であれば圧縮

とする。但し $UN1 > UN2 > UN3$ である。

このようにすることで、人気のないブログにおける不人気記事は、人気のあるブログにおける不人気記事より圧縮可と判定されやすくなる。

ここではユニークユーザ数を挙げたが、上述の他の指標値でも同様に用いることができる。

[0059] ステップS114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DB53に記憶された圧縮可／不可のフラグを更新又は維持する。

[0060] 以上で1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S115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S110に戻って次の記事を選択する。そしてステップS111～S114を実行する。

なお、第1指標値を用いたブログの人気度の判定は、当該ブログについて最初にステップS113を実行するときのみ行い、以後はステップS115からS120に進むときまで記憶しておけばよい。或いは、第1指標値によりブログの人気度を判定する処理はステップS102の直後に1回、行っても良い。

[0061] 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S120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S101に戻って、他の1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S120から図5の圧縮判定処理を終える。

[0062] 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DB51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DB53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。時期毎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。

[0063] 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。例えば全部又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S201で処理対象のブログを1つ特定する。

ステップS202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DB53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の圧縮判定処理で記憶された、記事毎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S203でブログサーバ1は、圧縮可とされている記事を圧縮対象の記事として特定する。

[0064] もし圧縮可とされている記事が当該ブログに存在しなければ、ブログサーバ1はステップS204からS210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S201に戻って、他の1つのブログを処理対象として特定する。

[0065] 圧縮対象として1以上の記事が存在した場合は、ブログサーバ1はステッ

プS 2 0 4 からS 2 0 5 に進み、記事圧縮を行う。即ち、ステップS 2 0 3 で特定した1又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。

[0066] 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

[0067] 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。

[0068] 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。

[0069] 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。

[0070] また圧縮対象部分の設定は、ブログ毎や記事毎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。

さらにブログ毎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。

[0071] 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。

[0072] 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に対して多様な圧縮方式が存在し、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良いし、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。

またブログ毎や記事毎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。

[0073] 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又は複数の記事について圧縮処理を行う。これに伴って圧縮処理を行った記事についての判定情報、つまり圧縮可とされていた情報（フラグ情報）をクリアする。

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。

- [0074] ステップS 2 0 7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。

圧縮処理を行うことによって、圧縮後の記事については、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。

このステップS 2 0 7でブログサーバ1、圧縮前の元の記事データから、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又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S 2 0 8で圧縮記事タグとして管理DB 53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又は複数の語句を登録する。

- [0075] 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良いため、このステップS 2 0 7、S 2 0 8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S 2 0 7、S 2 0 8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。

- [0076] ステップS 2 0 9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。

[0077] 以上の処理により1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S210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。

[0078] なお、図5の処理で圧縮可と判定された記事が図6の処理で圧縮されるものとしているが、図5の処理で設定される判定情報（例えばフラグ）は、適宜リセットされることが好適である。例えば圧縮判定処理の開始時にリセットする。最新の判定結果を反映して圧縮処理が行われるようにするためである。

[0079] 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。

ユーザ端末5からのアクセス要求があるとブログサーバ1はステップS301からS302に進み、まず要求された記事が、現在圧縮状態で保存されている記事（「圧縮記事」と表記する）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S302からS303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DB51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。

[0080] 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S304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DB51から読み出し、解凍処理を行う。そしてステップS305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。

- [0081] 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。

- [0082] 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、即ち圧縮状態から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、この記事が、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はか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。

- [0083] 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S305から図8BのステップS320に進む例が考えられる。ステップS320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S321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。

なお、解凍した記事データは、圧縮前の元の記事データと同一のデータである場合もあるが、不可逆圧縮を用いた場合は、元の記事データよりも低品質なデータとなる。つまり必ずしも元の記事データと同一ではない。そこで圧縮後に解凍した記事データについては「解凍記事」と表記する。

ステップS322では、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、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。

[0084] 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、不可逆圧縮を用いて圧縮処理を行った場合は、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、オリジナルの未圧縮の状態で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くなるという利点がある。

[0085] 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。

[0086] 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。

[0087] <4. 第2の実施の形態>

第2の実施の形態として、上述の図5の圧縮判定処理と同等の他の例を説明する。

これはブログサーバ1が、図5の圧縮判定処理に代えて図9Aの人気度判定処理及び図9Bは圧縮記事判定処理を行う例である。

[0088] 図9Aは、ブログ毎に人気度判定を行う処理である。

ブログサーバ1は図9Aの人気度判定処理を、逐次、ブログDBに保存している全部又は一部のブログに対して実行する。

ステップS101でブログサーバ1は、人気度判定処理の対象とする1つのブログを特定する。ステップS102でブログサーバ1は、処理対象としたブログについての第1指標値を取得する。例えば管理DB53から、当該ブログ（ブログID）について記憶されているブログ実績情報に含まれる第1指標値を読み出す。以上のステップS101、S102は図5と同様である。

[0089] ブログサーバ1はステップS103で、当該ブログについて第1指標値を

用いた人気度判定を行う。例えばページビュー数などに基づいて、人気度を複数段階のいずれかに分類する。

そしてステップS 1 0 4で、当該ブログについて、人気度判定情報を管理DB 5 3に登録する。例えば「人気／普通／不人気」のいずれかを示す値を判定情報の1つとして登録する。

[0090] ステップS 1 0 5でブログサーバ1は、続いて他のブログについても人気度判定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1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。

一方、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 0 5から図9 Aの人気度判定処理を終える。

[0091] ブログサーバ1は、この図9 Aの人気度判定処理を、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。これにより、各ブログについてのその時点の人気度判定情報が管理DB 5 3に記憶される。

[0092] またブログサーバ1は図9 Bの圧縮記事判定処理を逐次行う。

ステップS 1 3 0でブログサーバ1は、圧縮記事判定処理の対象とする1つのブログを特定する。

ステップS 1 3 1でブログサーバ1は、処理対象としたブログについての人気度判定情報を管理DB 5 3から取得する。

そして当該ブログの各記事についての圧縮判定として、ステップS 1 1 0～S 1 1 5の処理を実行する。これは図5のステップS 1 1 0～S 1 1 5と同様の処理である。

このときステップS 1 1 3の圧縮可否判定では、既に取得されている人気度判定情報を参照し、その人気度によって、アクセス可能性による圧縮可の判定基準を変更する。

これによってブログ自体の人気度を考慮したうえで、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。

[0093] 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 3 2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 3 0に戻って、他の1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 3 2から図9 Bの圧縮記事判定処理を終える。

[0094] 以上の図9 A、図9 Bの処理をブログサーバ1が逐次実行することで、逐次、圧縮すべき記事が判定される。圧縮可と判定された記事は、その後の図6の圧縮処理で圧縮されることになる。

この第2の実施の形態では、人気度判定処理と圧縮記事判定処理を、別個に行っていることで、処理のスケジュールの自由度を高めることができる。

例えば急激な人気上昇に対応するために、人気度判定処理のみを短い間隔で行いたい場合などに好適である。

[0095] <5. 第3の実施の形態>

第3の実施の形態として、ブログサーバ1が、第1指標値と第2指標値とに基づいて、記事について、圧縮するか否かの判定に加えて、既に圧縮した記事を解凍するか否かも判定する例を説明する。

[0096] ブログサーバ1は図10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10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良い。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。

[0097] ブログサーバ1は処理対象のブログを特定し(S 1 0 1)、第1指標値を取得(S 1 0 2)した後、ステップS 1 1 0、S 1 1 2、S 1 4 0、S 1 4 1、S 1 1 5で各記事について判定を行う。

この場合ブログサーバ1は、ステップS 1 1 0で1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 2に進んで第2指標値を取得する。そしてステップS 1 4 0では、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、未圧縮の記事もしくは解凍記事については、図5のステップS113と同様に、第1指標値と第2指標値を用いて圧縮可否判定を行う。

[0098] 一方、圧縮記事についてはブログサーバ1は、第1指標値と第2指標値を用いて解凍可否判定を行う。即ち、記事自体のアクセス可能性のみで解凍可否を判断せず、第1指標値によりブログの人気度を判定し、それを反映させた解凍可否判定を行う。

例えばまず人気度を、圧縮可否判定の場合と同様に、総ページビューなどの第1指標値を用いて、「人気／普通／不人気」の3段階などに判定する。

そして第2指標値として、例えば当該記事のページビュー数Nを用い、

- ・不人気ブログの場合・・・ページビュー数NがN1以上であれば解凍
- ・普通ブログの場合・・・ページビュー数NがN2以上であれば解凍
- ・人気ブログの場合・・・ページビュー数NがK3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。

[0099] 他の例として、第2指標値として、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、

- ・不人気ブログの場合・・・増加傾向値KがK1以上であれば解凍
- ・普通ブログの場合・・・増加傾向値KがK2以上であれば解凍
- ・人気ブログの場合・・・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。

[0100] そしてステップS 1 4 1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又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／不可のフラグ又は解凍可／不可のフラグを更新又は維持する。

ステップS 1 1 5, S 1 2 0は図5と同様である。

[0101] ブログサーバ1はまた、図11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。

[0102] 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 0 3で、圧縮可否の判定情報を参照して圧縮する記事を特定する。そしてステップS 2 0 4で圧縮する記事の有無により処理を分岐する。

1又は複数の記事が圧縮対象とされた場合、ブログサーバ1はステップS 2 0 5で、1又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。

[0103] 以上のステップS 2 0 5～S 2 0 9を行った場合、もしくはステップS 2 0 4で圧縮する記事が存在しないとされた場合は、ブログサーバ1はステップS 2 2 0で、解凍する記事を特定する。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を特定する。そしてステップS 2 2 1で解凍する記事の有無により処理を分岐する。解凍する記事が存在しなければステップS 2 2 4に進む。

[0104] 1又は複数の記事が解凍対象として特定された場合、ブログサーバ1はステップS 2 2 2で、当該特定された1又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S 2 2 3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。

[0105] ブログサーバ1はステップS 2 2 4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。さら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。

[0106] 以上の処理により1つのブログについての圧縮や解凍を終えたら、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11の圧縮解凍処理を終える。

以上の図11の圧縮解凍処理を行うことで、図10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。

[0107] 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。第3の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、圧縮状態から解凍しておく。

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。

[0108] ＜6. 第4の実施の形態＞

第4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、圧縮状態から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。

[0109] 図12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S401でブログサーバ1は1又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。

[0110] ブログサーバ1はステップS402で処理対象の1つのブログを特定する。そしてステップS403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良い。

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S407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S404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又は複数の圧

縮記事について設定されたタグ情報を含む。図6のステップS208で登録されたタグ情報である。

- [0111] ステップS405でブログサーバ1は、ステップS401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又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S406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。

- [0112] 以上の処理により1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S407で他のブログについて処理を行うか否かを確認する。行う場合はステップS402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S403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S407から図12の解凍判定処理を終える。

- [0113] この解凍判定処理とともに、ブログサーバ1は図13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S220A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を特定する。そしてステップS221Aで解凍する記事の有無により処理を分岐する。解凍する記事が存在しなければステップS210に進む。

- [0114] 1又は複数の記事が解凍対象として特定された場合、ブログサーバ1はステップS222Aで、当該特定された1又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S223A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Ｓ２２４Ａ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。

[0115] 以上の処理により１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で他のブログについて処理を行うか否かを確認する。処理を行う場合はステップＳ２０１に戻って、他のブログを特定し、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１３の解凍処理を終える。

[0116] 以上の図１３の処理により、図１２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とできる。

[0117] ＜７．第５の実施の形態＞

第５の実施の形態は、ブログの人気度の変化を監視し、或るブログについて人気度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。

[0118] この場合の解凍判定処理を図１４に示す。

ブログサーバ１はステップＳ４２０で処理対象の１つのブログを特定する。そしてステップＳ４２１で、管理ＤＢ５３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Ｓ４２６に進む。

[0119] 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S422に進み、当該ブログの実績情報を管理DB53から取得する。

例えばここでは第1指標値としての情報を取得するが、特には、期間毎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、期間毎、例えば1日毎や1週間毎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などは、人気の変動を表す指標となるため、ステップS422で取得する情報として好適である。

[0120] ステップS423でブログサーバ1は、上記の第1指標値から、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毎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」又は「人気下降傾向」と判定した場合はステップS424からS426に進み、「人気上昇傾向」と判定した場合はステップS425に進む。

[0121] ステップS425でブログサーバ1は、当該ブログに含まれる1又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DB53に記憶する。

[0122] 以上の処理により1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S426で他のブログについて処理を行うか否かを確認する。処理を行う場合はステップS420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S421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S426から図14の解凍判定処理を終える。

[0123] この解凍判定処理とともに、ブログサーバ1は図13の解凍処理又は図11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。

人気が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。

[0124] なお、図14のステップS425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。

[0125] <8. 第6の実施の形態>

第6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8B又は図8C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。

[0126] 図15に第6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。

[0127] この圧縮判定処理でも、ステップS101でブログを特定し、ステップS102で第1指標値を取得した後、各記事についてステップS110以降の処理を行う。

ブログサーバ1はまずステップS110で圧縮判定する記事を選択したら、ステップS150で、その記事が圧縮状態から解凍された記事（解凍記事）であるか否かを確認する。例えば管理DB53の圧縮解凍情報を参照して確認する。

解凍記事でなければステップS112以降にすすみ、図5と同様の処理を行う。

[0128] 一方、解凍記事であれば、ステップS151で、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期間経過していれば、ステップS112に進み、第2指標値を取得してステップS113で第1指標値及び第2指標値を用いた圧縮可否の判定を行う。

解凍から所定期間経過していなければ、圧縮可否判定を行わずにステップS115に進む。

[0129] 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。

[0130] なおステップS150の処理は、図7のようにアクセス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、ユーザ端末5からの閲覧のためのアクセス以外にも、例えばクローラによるアクセスもあり得る。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は人気度や閲覧のためのアクセス可能性の変化には関係ない。

そこでクローラのアクセスに応じて解凍したような場合は、ステップS150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。

さらには、クローラからのアクセスの場合、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。

[0131] <9. 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6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又は複数の記事が含まれるブログの人気度を示す第1指標値を取得する第1取得部12と、ブログに含まれる複数の記事ごとに当該記事に対するアクセス可能性を示す第2指標値を取得する第2取得部13と、ブログに含まれる記事について前記第1指標値と前記第2指標値とに基づいて圧縮するか否かを判定する判定部14とを備える。これらの機能により図5，図9A及び図9B，図10，図15の処理で圧縮可否の判定を行うようにしている。

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。そこでこれらの判定処理では、ブログに含まれる記事の内、アクセス可能性の小さい記事（例えば不人気記事）を圧縮するためにその可否、つまりアクセス可能性が小さい記事であるか否かを判定する。この場合に、アクセス可能性の小さい記事が、人気の高いブログ内の記事であるか、人気の低いブログ内の記事であるかも考慮して圧縮するか否かを判定する。

単にアクセス可能性が小さいということを示す指標だけでなく、ブログの人気度を示す指標も用いて圧縮するか否かを判定することで、なるべく人気の低いブログの中のアクセス可能性が低い記事を圧縮対象とし、一方でアクセス可能性が低い記事であっても人気の高いブログの記事であれば、圧縮対象となりにくいような判定が可能となる。

例えば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、

記事だけでなく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、圧縮した記事に対してアクセスがあった場合、ブログサーバ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、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。

[0132] 第1～第6の実施の形態のブログサーバ1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とともに、既に圧縮された記事に対してのアクセスが発生した場合に、当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や図11の処理で判定部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。

[0133] 第3の実施の形態では、第1指標値と第2指標値とに基づいて、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10参照）。

即ち一旦圧縮した記事についても、その後、定期的又は不定期に第1、第2指標値を取得して、アクセス可能性の変化を確認する。もし最新の状況としてアクセス可能性が高くなっていたら、解凍しておくように判定する。

一旦圧縮した記事であっても、アクセス可能性が高くなった記事を解凍しておくようにすることで、アクセスが発生したときの解凍処理を不要とし、処理負担を削減し、またユーザに快適な閲覧を提供することができる。

例えばあるブログの記事について、その記事の著者によって内容の変更があったり、その記事に対して読者からコメントが投稿されたりした場合など、その記事自体や関連内容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。このような記事は、圧縮を解いておくことで、アクセス時に解凍しなくてもよいようにしている。

[0134] 第4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12参照）。

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、圧縮状態から解凍しておく。

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。

[0135] 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、また記憶リソースをむやみに圧迫しないようにできる。

[0136] 第5の実施の形態では、ブログの人気度の変化を監視し、或るブログについて人気度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14参照）。

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。

[0137] なお第5の実施の形態の図14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。

[0138] 第6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 15 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、次のアクセス時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、第 1, 第 2 指標値に基づいた判定で、アクセス可能性が低ければ圧縮すればよい。これにより、解凍したままとしておくことが無駄なリソース消費となっている場合に対処でき、必要な記憶容量を低減させ、記憶リソースの圧迫を回避する。

[0139] 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、最も人気度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（人気大）から第 $(n-1)$ レベル（人気低）については第 1 指標値及び第 2 指標値を用いて記事毎の圧縮可否判定を行うが、第 n レベル（人気最低）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。

[0140] 圧縮処理については段階的に複数回行うようにしても良い。

例えば図 10 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、さらに所定期間、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。

[0141] 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 10、図 15 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 10、図 12、図 14 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。

[0142] ブログの人気度の判定については、複数の第 1 指標値を用いてもよい。

同様に、記事のアクセス可能性の判定については、複数の第 2 指標値を用いてもよい。

[0143] 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を、頻繁に使用されるフォルダであるか否かという点でフォルダの人気度をはかり、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。

[0144] 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よい。

[0145] <10. 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第1取得部12、第2取得部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。

[0146] 実施の形態のプログラムは、複数の記事が含まれるブログの人気度を示す第1指標値を取得する手順（S102）と、ブログに含まれる複数の記事ごとに、当該記事に対するアクセス可能性を示す第2指標値を取得する手順（S112）と、ブログに含まれる記事について第1指標値と第2指標値とに基づいて圧縮するか否かを判定する手順（S113又はS140）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10、図15等で説明した処理を実行させるプログラムである。

[0147] 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又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

[0148] 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 第1取得部、13 第2取得部、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又は複数の記事が含まれるブログの人気度を示す第1指標値を取得する第1取得部と、
- 前記ブログに含まれる複数の記事ごとに、当該記事に対するアクセス可能性を示す第2指標値を取得する第2取得部と、
- ブログに含まれる記事について、前記第1指標値と前記第2指標値とに基づいて、圧縮するか否かを判定する判定部と、を備えた情報処理装置。
- [請求項2] 前記判定部によって圧縮すると判定された記事について圧縮を行い、また既に圧縮された記事に対してのアクセス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、前記第1指標値と前記第2指標値とに基づいて、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4] 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5] 前記判定部は、ブログの人気度の変化を監視し、或るブログについて人気度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の全部又は一部を解凍する記事と判定する
- 請求項1に記載の情報処理装置。
- [請求項6] 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、前記判定部は、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
- 請求項1に記載の情報処理装置。
- [請求項7] 前記第1指標値は、ブログ全体について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、ブログに設定された総被リンク

数、ブログに記述された総コメント数、ブログのページランクを示す値、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、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] 前記第 2 指標値は、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、記事に設定された被リンク数、記事のページランクを示す値、記事のページビューの増加傾向を示す値、記事のアクセスのない期間の長さを示す値、特定語句の有無を示す値、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9] 情報処理装置が実行する情報処理方法として、
1 又は複数の記事が含まれるブログの人気度を示す第 1 指標値を取得するステップと、
前記ブログに含まれる複数の記事ごとに、当該記事に対するアクセス可能性を示す第 2 指標値を取得するステップと、
ブログに含まれる記事について、前記第 1 指標値と前記第 2 指標値とに基づいて、圧縮するか否かを判定するステップと、を行う
情報処理方法。

[請求項10] 1 又は複数の記事が含まれるブログの人気度を示す第 1 指標値を取得する手順と、
前記ブログに含まれる複数の記事ごとに、当該記事に対するアクセス可能性を示す第 2 指標値を取得する手順と、
ブログに含まれる記事について、前記第 1 指標値と前記第 2 指標値とに基づいて、圧縮するか否かを判定する手順と、
を情報処理装置に実行させるプログラム。

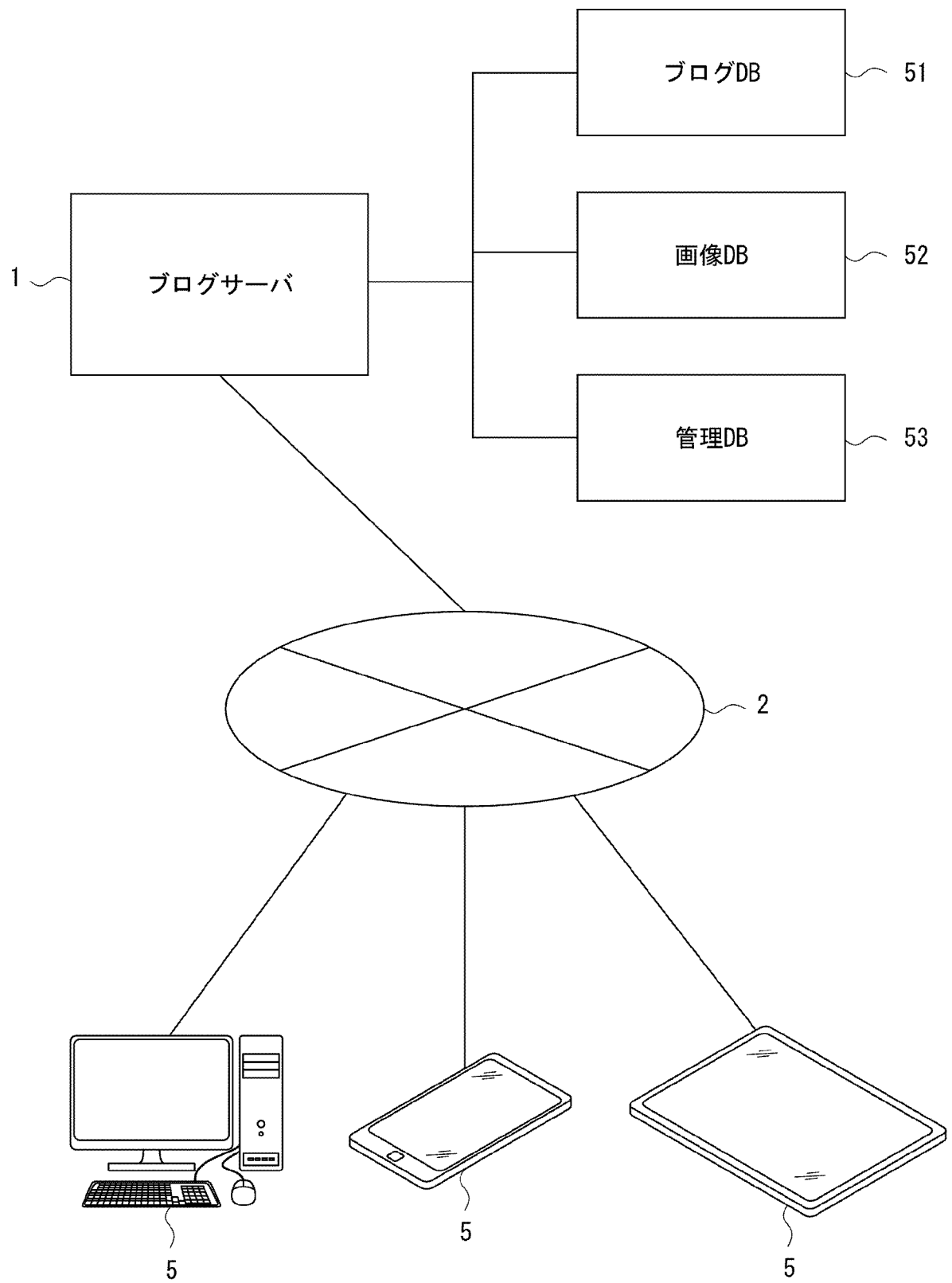
[請求項11] 1 又は複数の記事が含まれるブログの人気度を示す第 1 指標値を取得する手順と、

前記ブログに含まれる複数の記事ごとに、当該記事に対するアクセス可能性を示す第2指標値を取得する手順と、

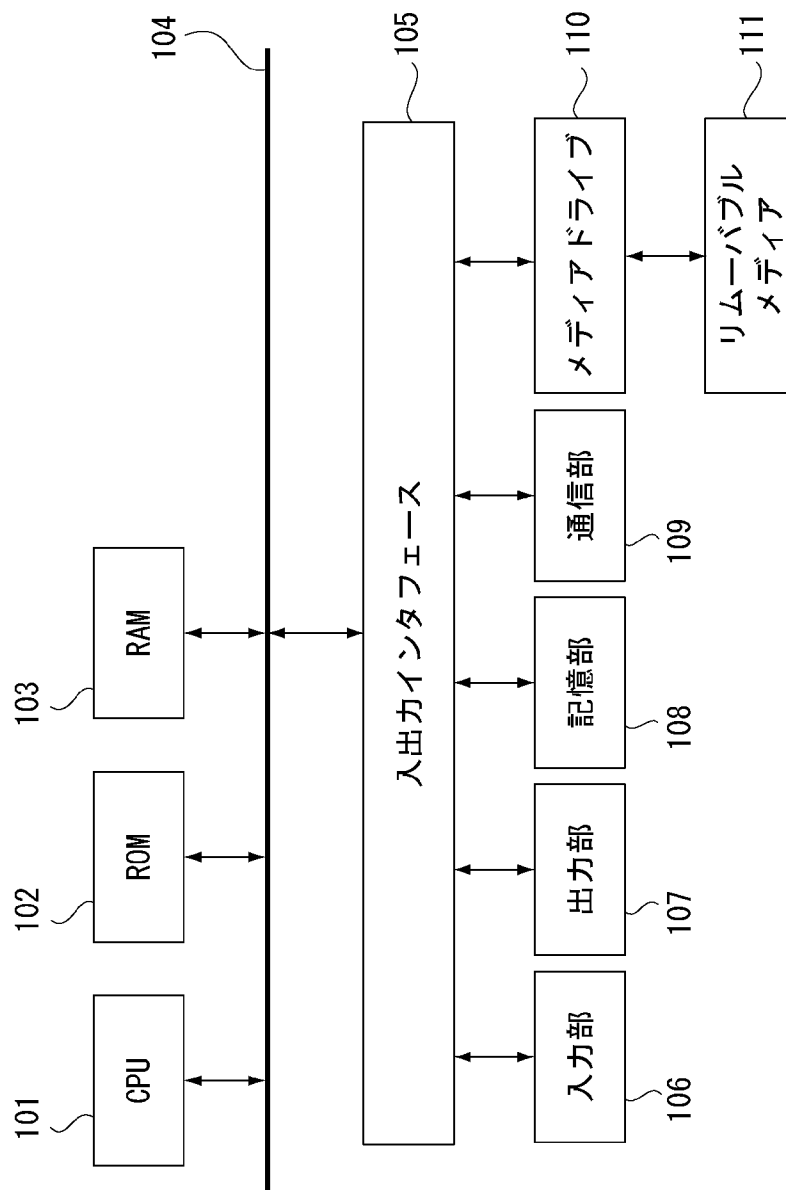
ブログに含まれる記事について、前記第1指標値と前記第2指標値とに基づいて、圧縮するか否かを判定する手順と、

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。

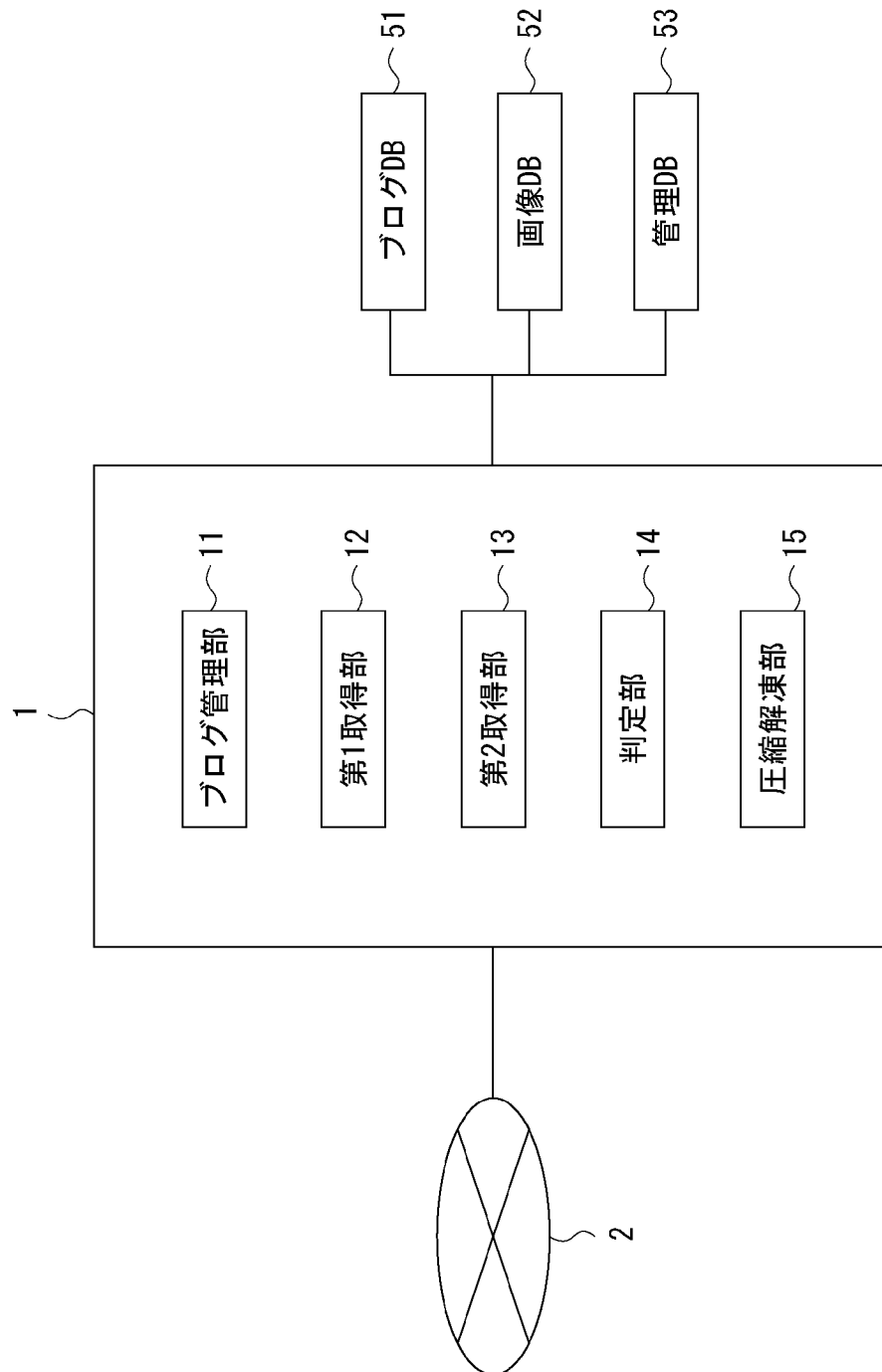
[図1]



[図2]



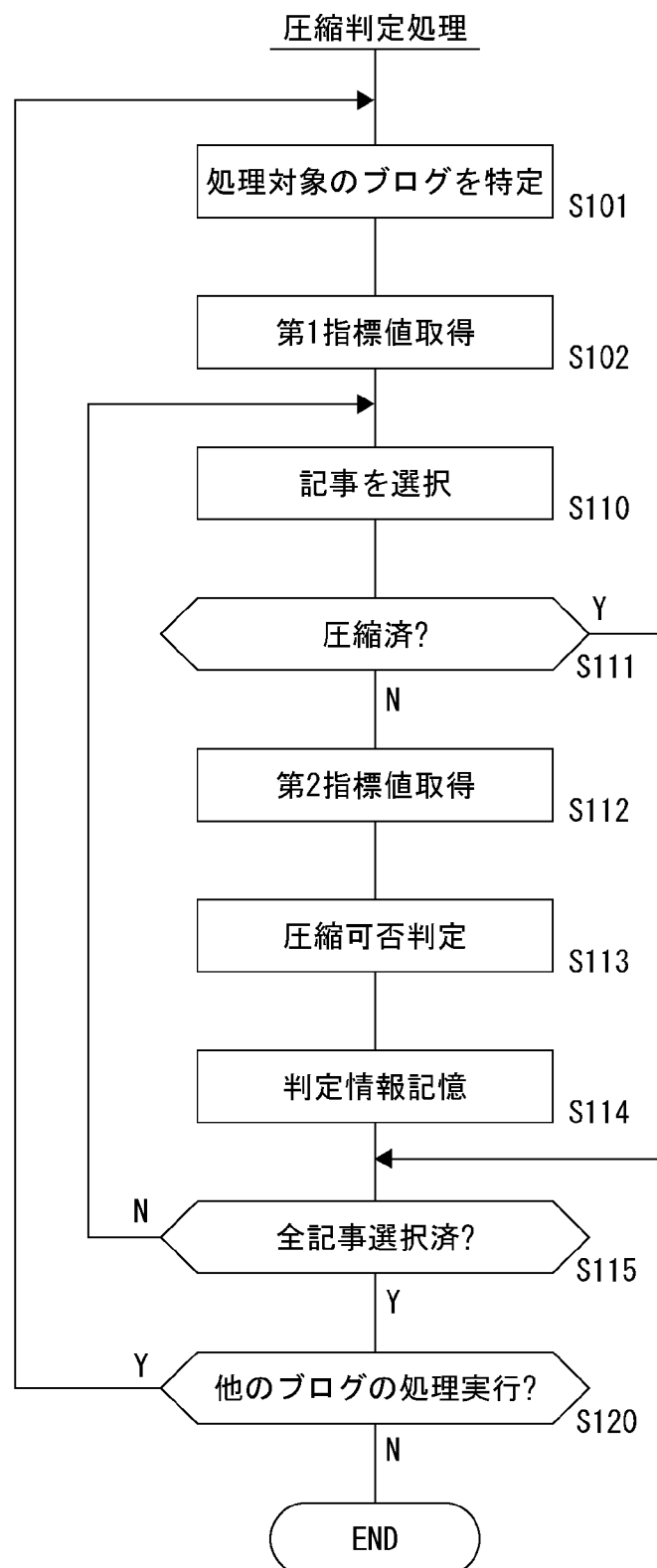
[図3]



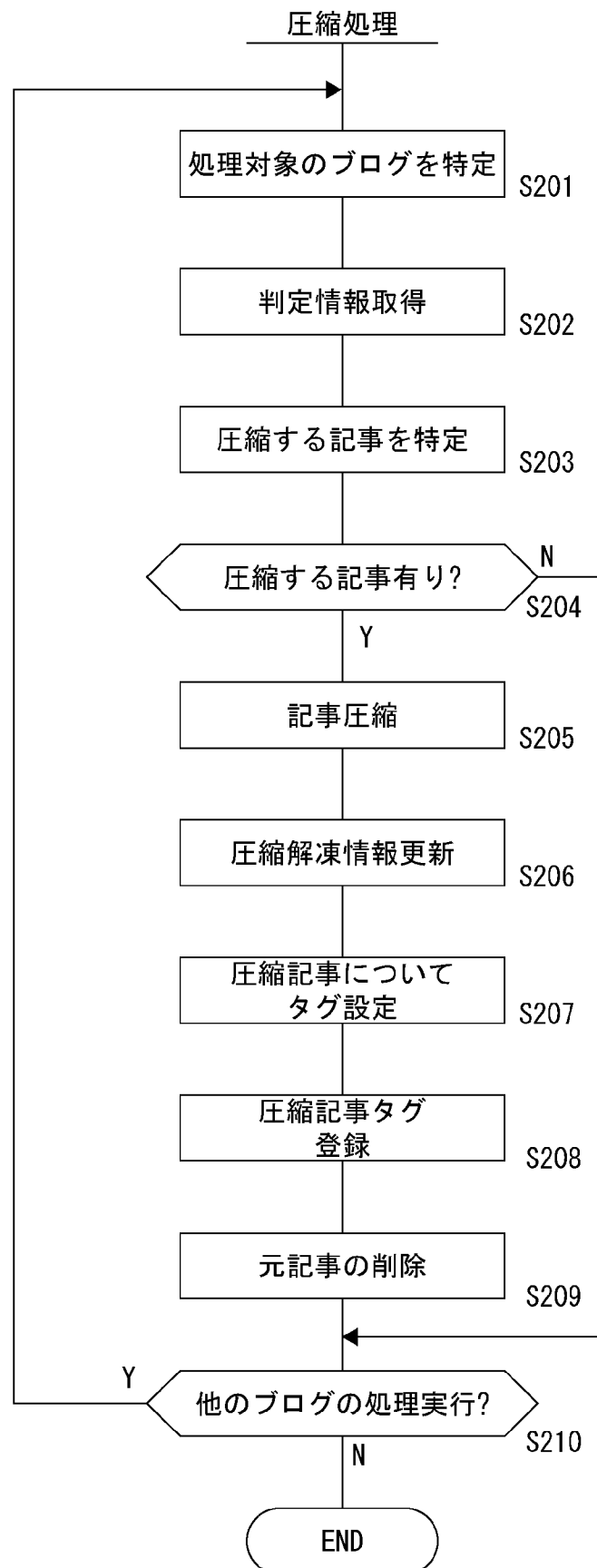
管理DB 53

[illegible]

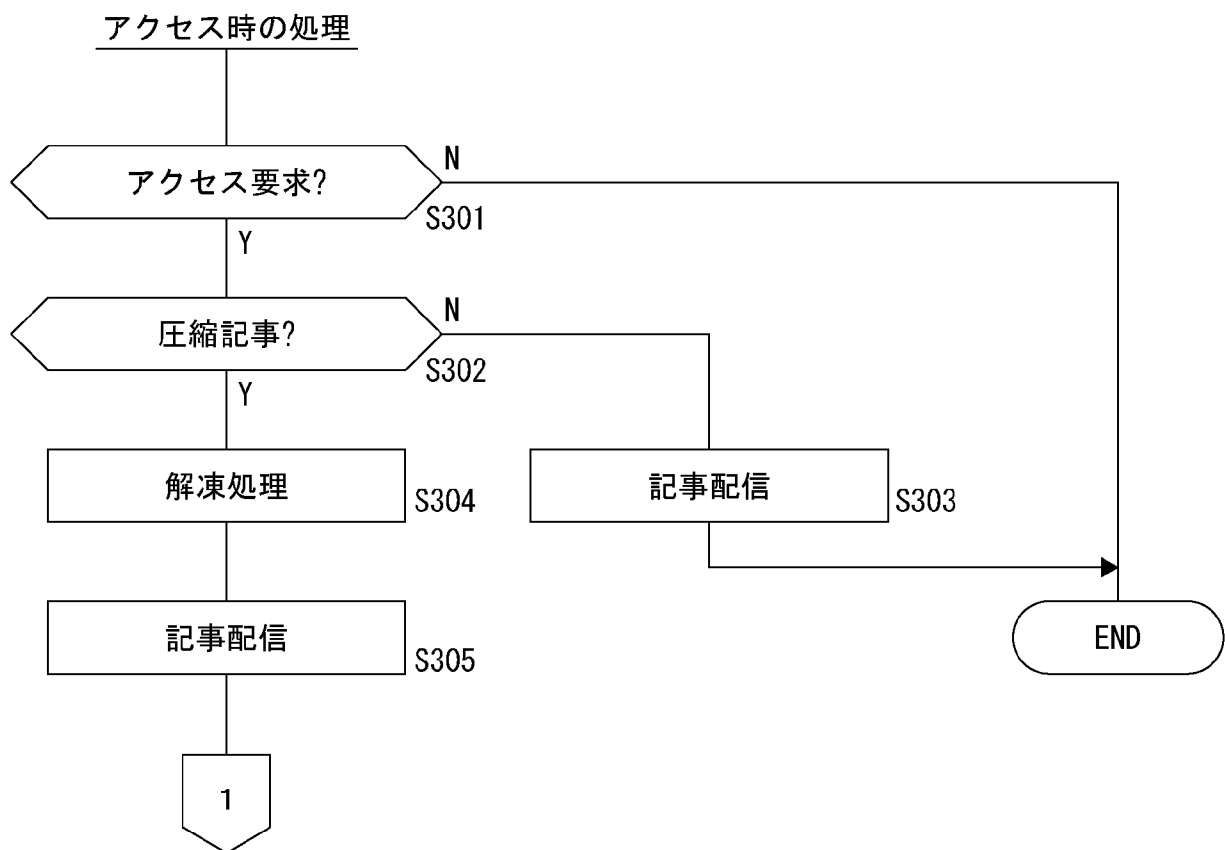
[図5]



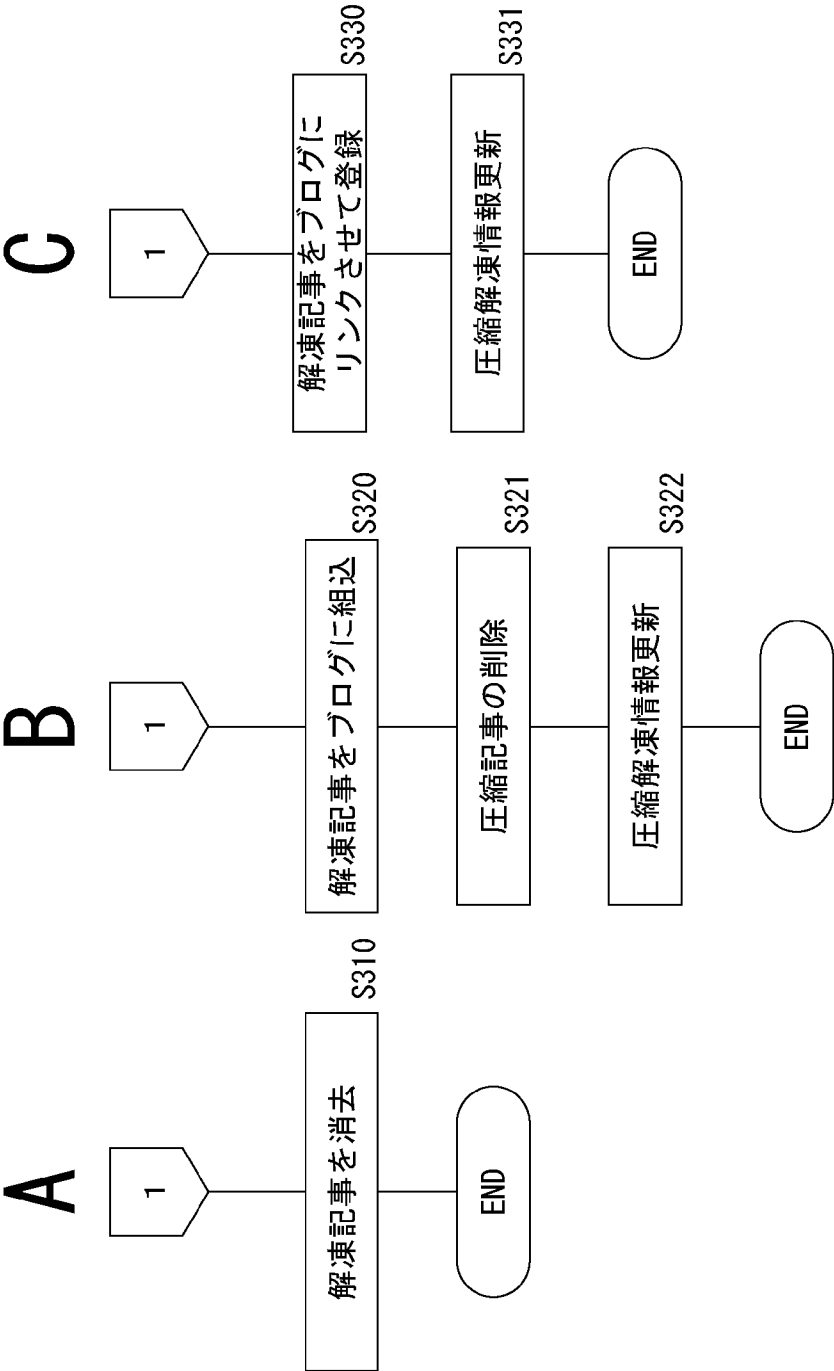
[図6]



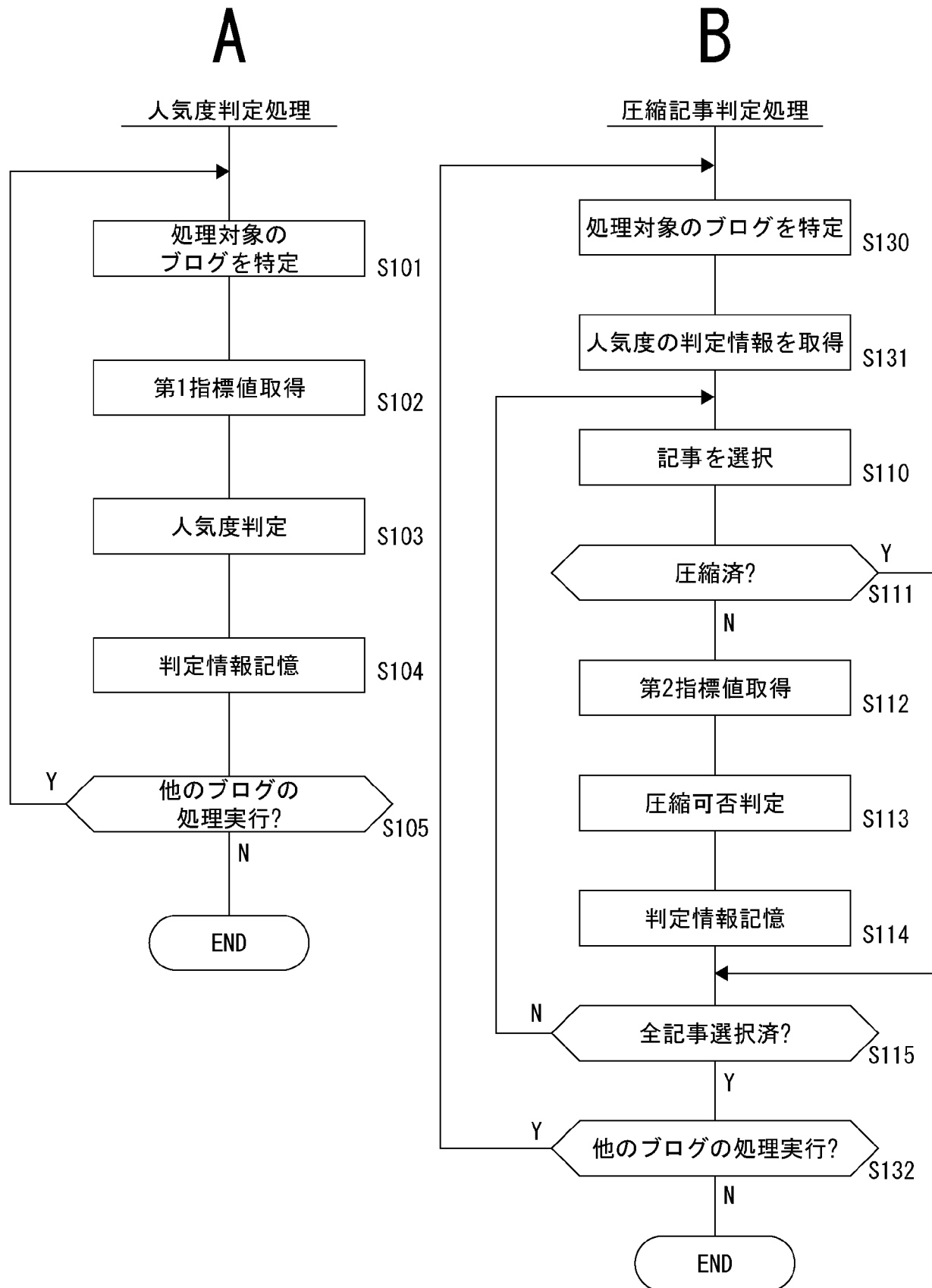
[図7]



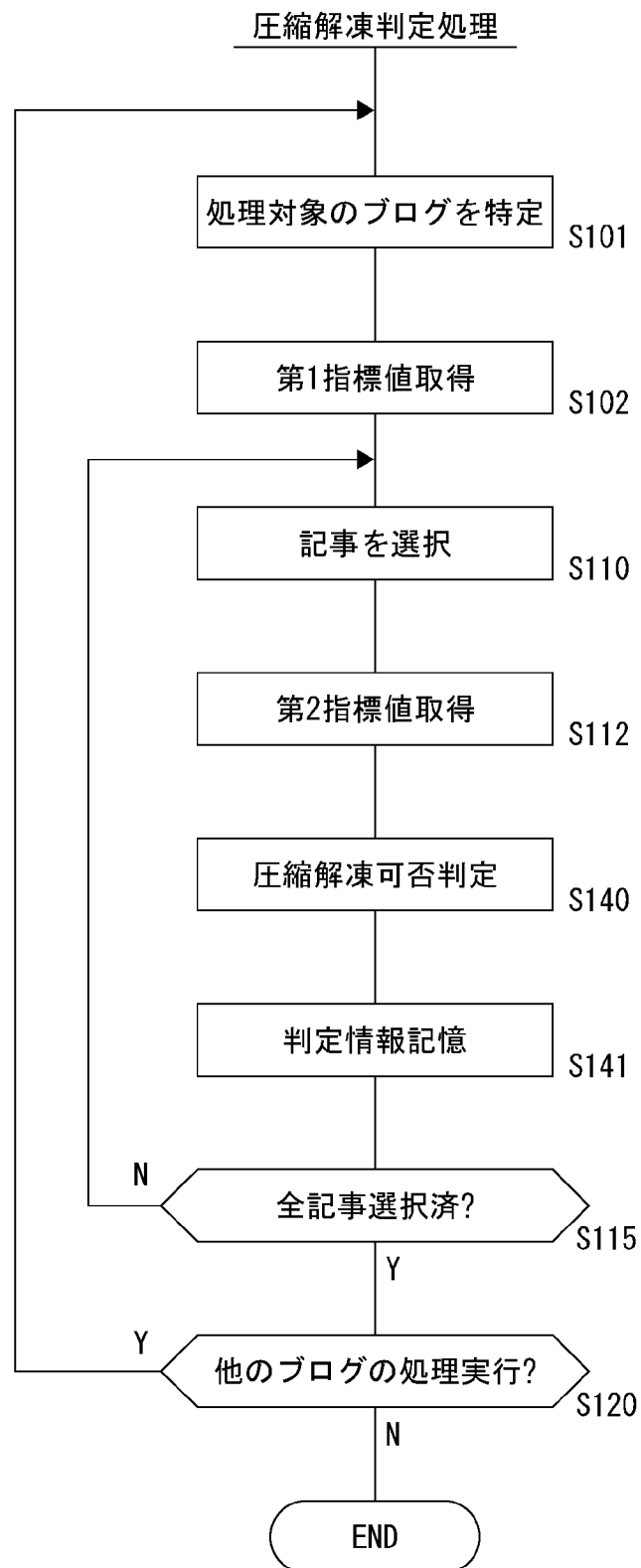
[図8]



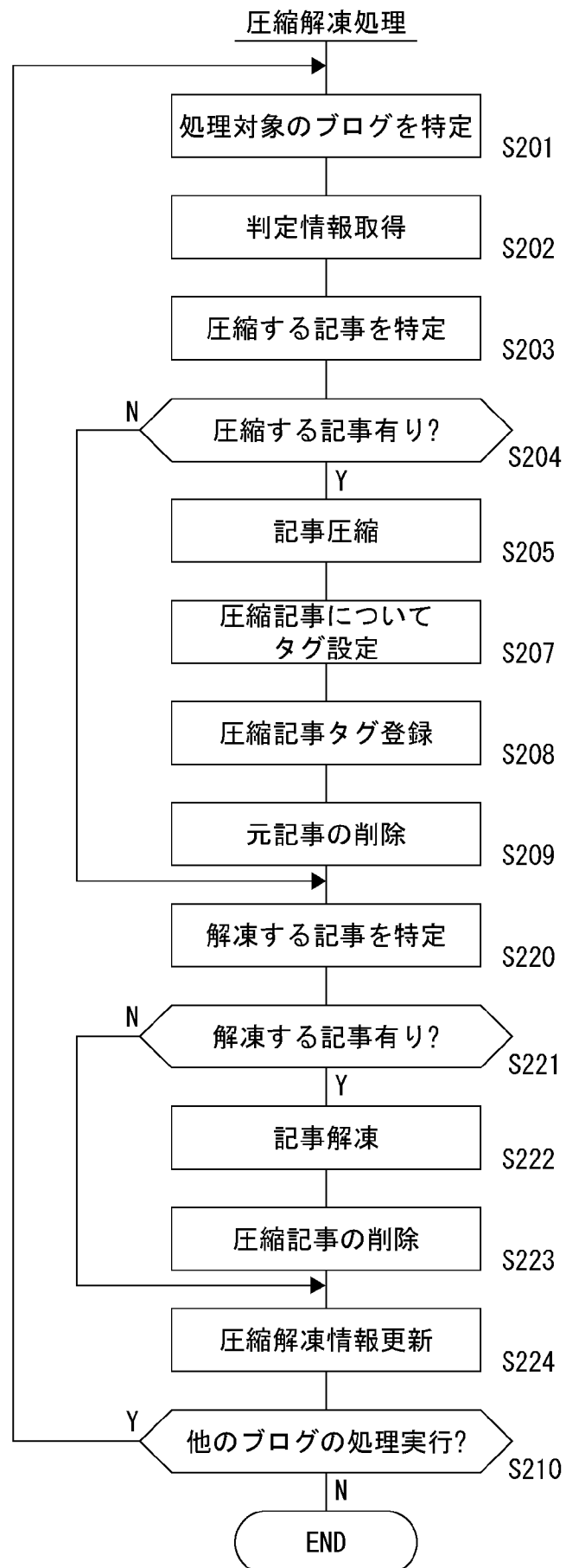
[図9]



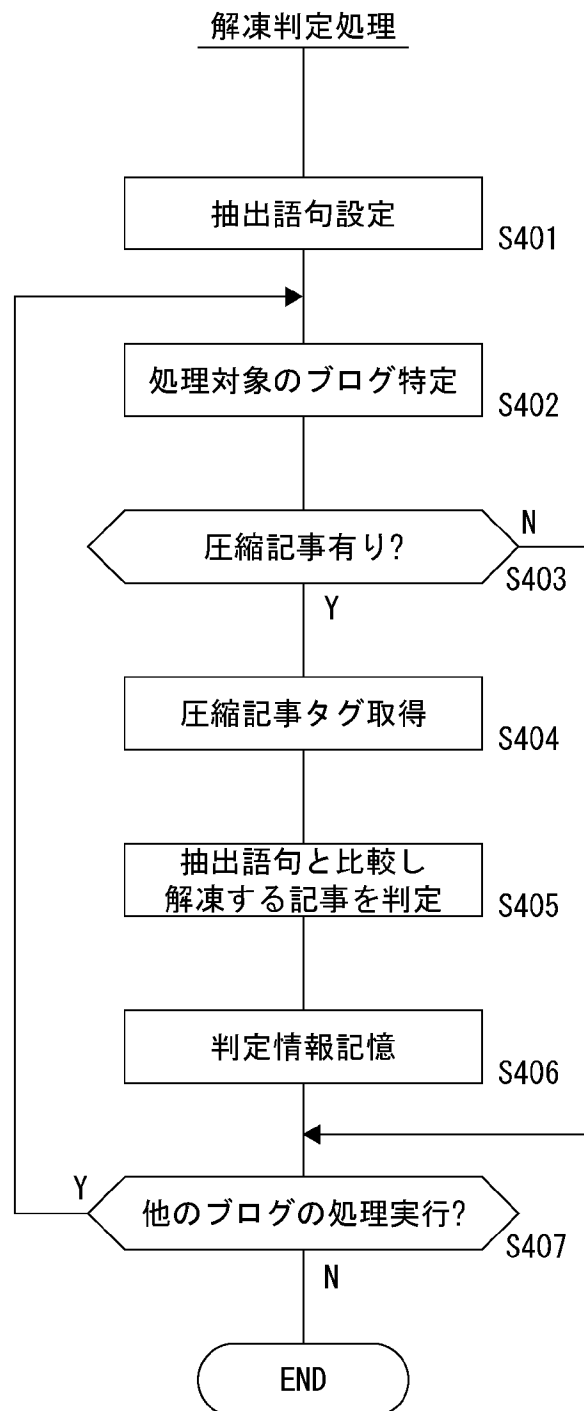
[図10]



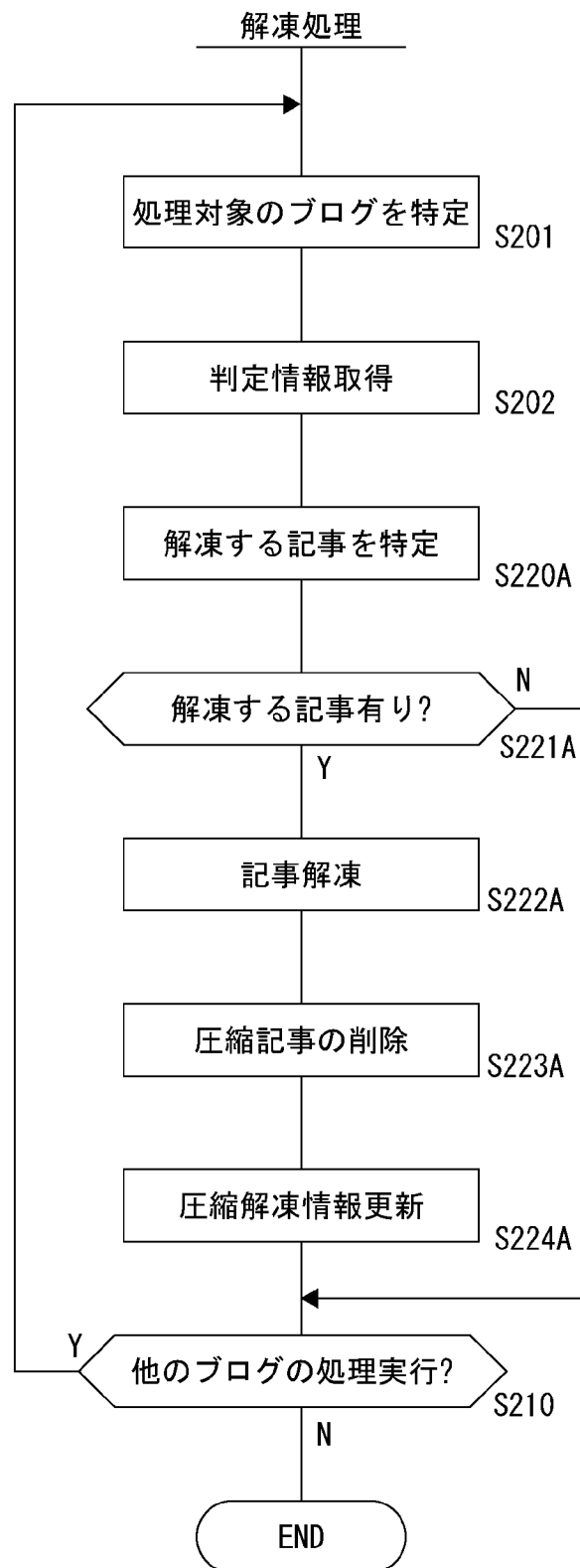
[図11]



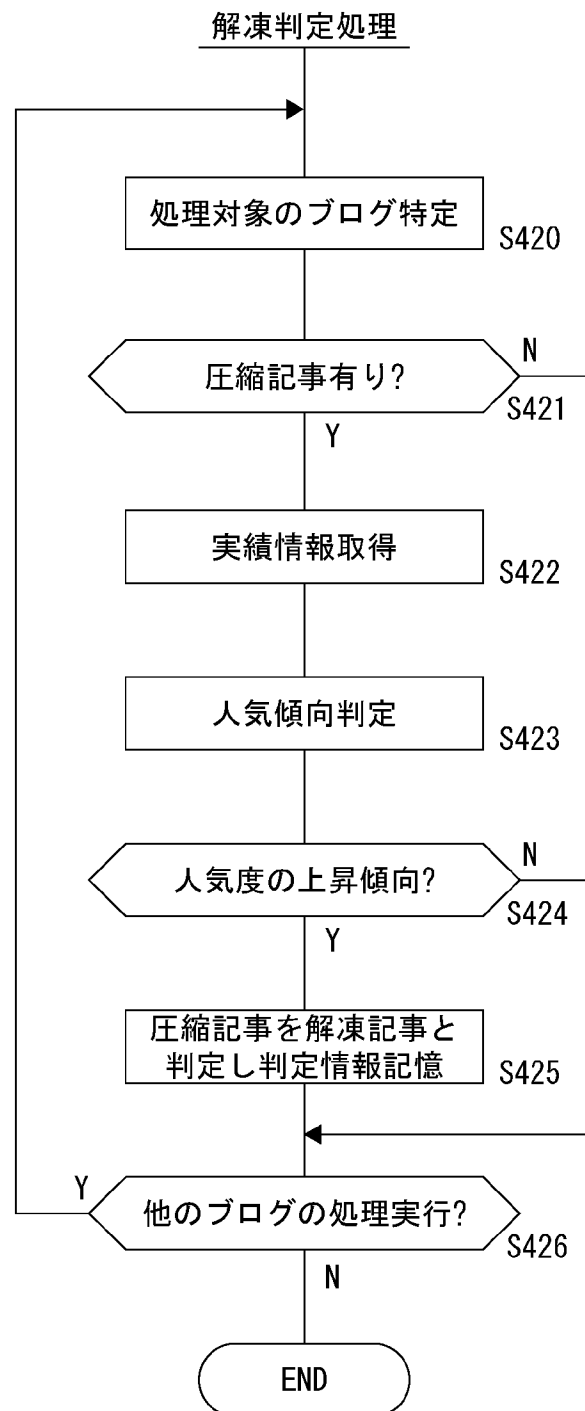
[図12]



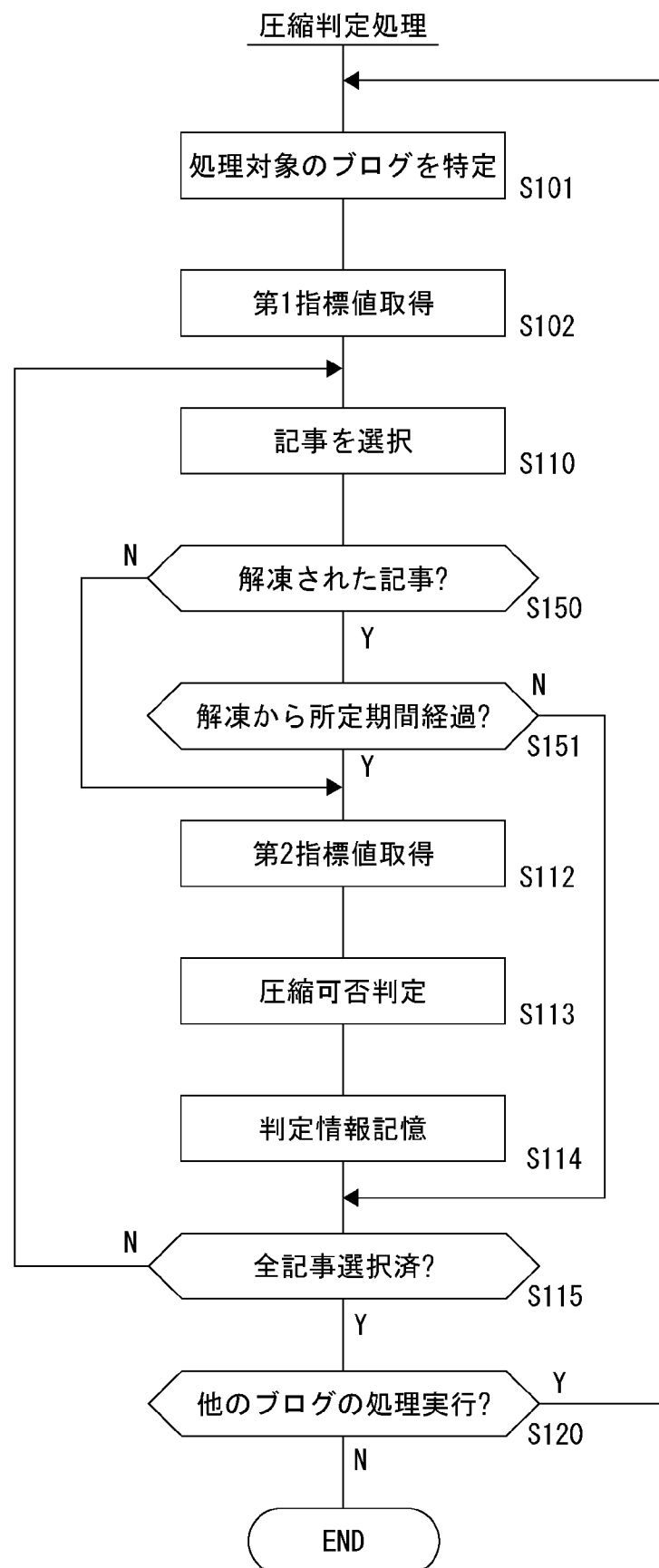
[図13]



[図14]



[図15]



INTERNATIONAL SEARCH REPORT

International application No.

PCT/JP2016/081143

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	1-11
A	US 2011/0138270 A1 (INTERNATIONAL BUSINESS MACHINES CORP.), 09 June 2011 (09.06.2011), & CN 102053990 A	1-11
A	JP 2012-222504 A (Sony Corp.), 12 November 2012 (12.11.2012), & US 2012/0257872 A1 & CN 102740033 A	1-11
A	JP 2012-14236 A (NTT Docomo Inc.), 19 January 2012 (19.01.2012), (Family: none)	1-11

☐ 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	1-11
A	US 2011/0138270 A1 (INTERNATIONAL BUSINESS MACHINES CORPORATION) 2011.06.09 & CN 102053990 A	1-11
A	JP 2012-222504 A (ソニー株式会社) 2012.11.12 & US 2012/0257872 A1 & CN 102740033 A	1-11

☒ 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)	1-11