

(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
2014 年 3 月 6 日(06.03.2014)



(10) 国際公開番号
WO 2014/034852 A1

- (51) 国際特許分類:
G06F 12/00 (2006.01) **G06F 17/30** (2006.01)
- (21) 国際出願番号: PCT/JP2013/073328
- (22) 国際出願日: 2013 年 8 月 30 日(30.08.2013)
- (25) 国際出願の言語: 日本語
- (26) 国際公開の言語: 日本語
- (30) 優先権データ:
特願 2012-192453 2012 年 8 月 31 日(31.08.2012) JP
- (71) 出願人: 日本電気株式会社(NEC CORPORATION)
[JP/JP]; 〒1088001 東京都港区芝五丁目 7 番 1 号
Tokyo (JP).
- (72) 発明者: 荒木 拓也(ARAKI Takuya); 〒1088001 東京都港区芝五丁目 7 番 1 号 日本電気株式会社
Tokyo (JP).
- (74) 代理人: 速水 進治(HAYAMI Shinji); 〒1410031 東京都品川区西五反田 7 丁目 9 番 2 号 五反田 T
Gビル 9 階 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, DK, DM, DO, DZ, EC, EE, EG, ES, FI, GB, GD, GE, GH, GM, GT, HN, HR, HU, ID, IL, IN, IS, JP, KE, KG, KN, KP, KR, KZ, LA, LC, LK, LR, LS, LT, LU, LY, MA, MD, ME, MG, MK, MN, MW, MX, MY, MZ, NA, NG, NI, NO, NZ, OM, PA, PE, PG, PH, PL, PT, QA, RO, RS, RU, RW, 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.

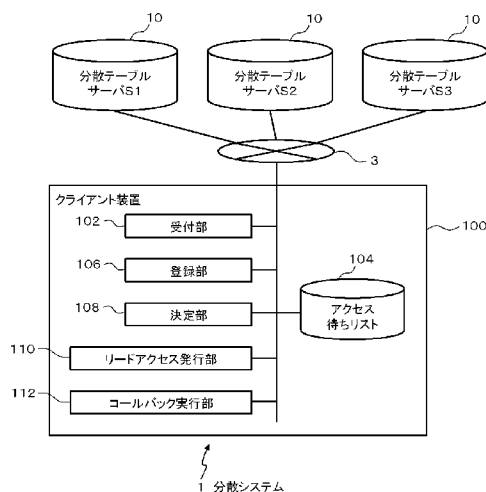
- (84) 指定国 (表示のない限り、全ての種類の広域保護が可能): ARIPO (BW, GH, GM, KE, LR, LS, MW, MZ, NA, RW, SD, SL, SZ, TZ, UG, ZM, ZW), ユーラシア (AM, AZ, BY, KG, KZ, RU, TJ, TM), ヨーロッパ (AL, AT, BE, BG, CH, CY, CZ, DE, DK, EE, ES, FI, FR, GB, GR, HR, HU, IE, IS, IT, LT, LU, LV, MC, MK, MT, NL, NO, PL, PT, RO, RS, SE, SI, SK, SM, 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:

— 国際調査報告 (条約第 21 条(3))

(54) Title: DISTRIBUTED SYSTEM, INFORMATION PROCESSING DEVICE DATA PROCESSING METHOD, AND PROGRAM

(54) 発明の名称: 分散システム、情報処理装置のデータ処理方法およびプログラム



- 1 Distributed system
10 Distributed table server (S1, S2, S3)
100 Client device
102 Receiving unit
104 Access wait list
106 Registration unit
108 Determination unit
110 Read access issuing unit
112 Call-back execution unit

(57) Abstract: A distributed system comprises: a receiving unit (102) which receives read access process keys which read out data from a plurality of distribution tables which are possessed by a plurality of servers (10), and a call-back which determines a process which is executed using a value which is obtained from the keys; a registration unit (106) which registers in an access wait list (104) the read access process keys and the call-back; a determination unit (108) which determines issuance destinations and timings of standby read accesses, according to a prescribed condition; a read access issuing unit (110) which issues the standby read accesses to the determined destination servers at the determined timing and receives results of the accesses; and a call-back execution unit (112) which, using the results of the accesses, executes the call-back which is registered in the access wait list (104).

(57) 要約: 分散システム (1) は、複数のサーバ (10) が有する複数の分散テーブルからデータを読み出すリードアクセス処理のキー、および当該キーから得たバリューを用いて実行する処理を定めるコールバック、を受け付ける受付部 (102) と、リードアクセス処理のキーとコールバックをアクセス待ちリスト (104) に登録する登録部 (106) と、所定の条件に従い、待機リードアクセスの発行先およびタイミングを決定する決定部 (108) と、決定した発行先のサーバに、決定したタイミングで、待機リードアクセスを発行し、アクセス結果を受け取るリードアクセス発行部 (110) と、アクセス結果を用いて、アクセス待ちリスト (104) に登録されたコールバックを実行するコールバック実行部 (112) と、を備える。

明 細 書

発明の名称：

分散システム、情報処理装置のデータ処理方法およびプログラム

技術分野

[0001] 本発明は、分散システム、情報処理装置のデータ処理方法およびプログラムに関する。

背景技術

[0002] 近年、大規模なデータを対象にした機械学習やデータマイニング等の分析が行われるようになってきている。大規模なデータを扱うためのフレームワークとして、MapReduce（マプリデュース）がGoogle（グーグル）社によって提案され、そのオープンソース実装であるHadoop（ハドゥープ）の利用がデータ分析に用いられるようになってきている。

[0003] MapReduceは、次のように動作する。まず、MapReduceの入力として分散ファイルシステム上のファイルを用いる場合を想定する。この場合、ファイルは分割され、その断片が複数のマシン上に存在する。MapReduceでは、まずこの各ファイル断片に対して分散して「map」関数を実行する。「map」関数の出力はKey-Value（キーバリュー）のペアである。システムは複数のマシンから生成されるKey-Valueペアをキー（Key）でグループ化する。そのため、同じキーを持つKey-Valueペアが同じマシンに転送されるよう、通信が行われる。そして、これを入力として「reduce」関数を実行する。これを図25に示す。

[0004] MapReduceを用いることで、様々なアルゴリズムが記述できることが知られているが、場合によっては低速になることも知られている。低速になるケースの一例に、map関数、あるいはreduce関数から、複数のマシンで共有されるデータにランダムアクセスする必要があるような場合がある。

[0005] このような共有データへのランダムアクセス処理が必要となる例として、

機械学習における「モデル」の更新がある。モデルは学習結果を表すデータで、入力データ（教師データ）により更新される。そして、もっともらしいモデルデータが得られるまで、入力データによってモデルデータの更新を何度も行う。ここで、モデルデータが更新される場所はランダムである。

[0006] MapReduceでは複数のマシンで共有されるデータ構造を直接はサポートしない。MapReduceが利用する分散ファイルシステムを使うことも考えられるが、ランダムアクセスは非常に低速であり、実用的ではない。

[0007] この問題の解決を目指したシステムの一例が、非特許文献1に記載されている。このシステムでは、複数のマシンで共有されるデータとして、汎用の分散型メモリキャッシュシステムであるmemcachedを用い、Hadoopと組み合わせて用いることでこの課題を解決しようとしている。ここで、memcachedはメモリ上にKey-Valueペアを保存し、キーからバリューを引けるようにしたデータストアである。複数のマシン上でmemcachedを動作させ、各マシンが担当するキーの範囲を決めておくことで、複数のマシン上に分散したオンメモリのデータストアを実現する。

[0008] 非特許文献1では、map関数やreduce関数から、これら複数のマシン上のmemcachedにアクセスすることで、前記の課題を解決しようとしている。これを図26に示す。

また、特許文献1には、MapReduceを実装したオープンソースソフトウェアであるHadoop等を用いて実現される分散処理システムの例が記載されている。

先行技術文献

特許文献

[0009] 特許文献1：特開2012-58836号公報

非特許文献

[0010] 非特許文献1：Jimmy Lin、外3名、“Low-Latency, High-Throughput Access to Static Global Resources within the Hadoop Framework”、[online]

、2009年、HCIL Technical Report HCIL-2009-01、[平成24年5月25日検索]、インターネット〈URL : <http://hcil.cs.umd.edu/trs/2009-01/2009-01.pdf>〉

発明の概要

発明が解決しようとする課題

[0011] 非特許文献1ではm e m c a c h e dを用いているが、以降、一般的な呼称として「分散テーブル」と呼ぶものとする。

非特許文献1に示されている方式では、分散テーブル上のデータにアクセスする際のレイテンシが大きいという課題があった。ネットワークを経由して他のマシン上のバリューを取得する場合、たとえば1 m sといったような長い時間がかかる。この場合、1秒間に1000回しかデータにアクセスできず、このデータアクセスが全体の実行におけるボトルネックとなる。

[0012] これを改善するためには、できるだけ複数のデータを一度に読み書きすればよい。すなわち、たとえばKey1とKey2に対応するバリューを取得したい場合、Key1とKey2が同じマシンの担当になっているのであれば、Key1のバリューを取得した後に、Key2のバリューを取得しに行くのではなく、Key1とKey2の両方のバリューを当該マシンに一度に要求する。こうすれば、2つのデータに対するアクセスが一度に行なえるため、レイテンシの影響を削減できる。

[0013] このような改善手法は非特許文献1においても示唆されているが、これをどのように実現すれば良いかについては述べられていない。

すなわち、上述した非特許文献1に記載のシステムにおいては、プログラム上同じ場所に現れる複数のアクセスを1つにまとめることは容易であるが、プログラム上別の場所に現れる、あるいはM a p R e d u c e プログラムにおいて、(同一マシン上で動作する)複数のm a p 関数にまたがるアクセスを1つにまとめるのは容易ではないという問題点があった。

[0014] 本発明の目的は、上述した課題である分散テーブルへのアクセス時のレイテンシの影響による処理の遅延を解決する分散システム、情報処理装置のデ

ータ処理方法およびプログラムを提供することにある。

課題を解決するための手段

[0015] 本発明の分散システムは、

複数のサーバがそれぞれ有する複数の分散テーブルからデータを読み出すリードアクセス処理のキー、および、前記キーに基づき読み出した結果を用いて実行する処理を定めるコールバック、を受け付ける受付手段と、

受け付けた前記リードアクセス処理を待機させ、該当するサーバに後で発行する待機リードアクセスとして前記キーと前記コールバックをアクセス待ちリストに登録する登録手段と、

前記アクセス待ちリストに登録された前記待機リードアクセスについて、所定の条件に従い、前記待機リードアクセスを発行する発行先のサーバおよび発行タイミングを決定する決定手段と、

前記決定手段が決定した前記発行先の前記サーバに、決定した前記発行タイミングで、前記待機リードアクセスを発行し、前記サーバから当該アクセス結果を受け取るリードアクセス発行手段と、

前記リードアクセス発行手段が受け取った前記アクセス結果を用いて、前記アクセス待ちリストに登録された該当する待機リードアクセスの前記コールバックを実行し、実行した前記コールバックの前記待機リードアクセスを前記アクセス待ちリストから削除するコールバック実行手段と、を備える。

[0016] 本発明の情報処理装置のデータ処理方法は、

情報処理装置が、

複数のサーバがそれぞれ有する複数の分散テーブルからデータを読み出すリードアクセス処理のキー、および、前記キーに基づき読み出した結果を用いて実行する処理を定めるコールバック、を受け付け、

受け付けた前記リードアクセス処理を待機させ、該当するサーバに後で発行する待機リードアクセスとして前記キーと前記コールバックをアクセス待ちリストに登録し、

前記アクセス待ちリストに登録された前記リードアクセスについて、所定

の条件に従い、前記待機リードアクセスを発行するサーバおよび発行タイミングを決定し、

決定した前記サーバに、決定した前記発行タイミングで、前記待機リードアクセスを発行し、前記サーバから当該アクセス結果を受け取り、

受け取った前記アクセス結果を用いて、前記アクセス待ちリストに登録された該当する待機リードアクセスの前記コールバックを実行し、実行した前記コールバックの前記待機リードアクセスを前記アクセス待ちリストから削除する情報処理装置のデータ処理方法である。

[0017] 本発明のコンピュータプログラムは、

情報処理装置を実現するコンピュータに、

複数のサーバがそれぞれ有する複数の分散テーブルからデータを読み出すリードアクセス処理のキー、および、前記キーに基づき読み出した結果を用いて実行する処理を定めるコールバック、を受け付ける手順、

受け付けた前記リードアクセス処理を待機させ、該当するサーバに後で発行する待機リードアクセスとして前記キーと前記コールバックをアクセス待ちリストに登録する手順、

前記アクセス待ちリストに登録された前記待機リードアクセスについて、所定の条件に従い、前記待機リードアクセスを発行するサーバおよび発行タイミングを決定する手順、

決定した前記サーバに、決定した前記発行タイミングで、前記待機リードアクセスを発行し、前記サーバから当該アクセス結果を受け取る手順、

受け取った前記アクセス結果を用いて、前記アクセス待ちリストに登録された該当する待機リードアクセスの前記コールバックを実行し、実行した前記コールバックの前記待機リードアクセスを前記アクセス待ちリストから削除する手順、を実行させるためのプログラムである。

[0018] なお、以上の構成要素の任意の組合せ、本発明の表現を方法、装置、システム、記録媒体、コンピュータプログラムなどの間で変換したものもまた、本発明の態様として有効である。

[0019] また、本発明の各種の構成要素は、必ずしも個々に独立した存在である必要はなく、複数の構成要素が一個の部材として形成されていること、一つの構成要素が複数の部材で形成されていること、ある構成要素が他の構成要素の一部であること、ある構成要素の一部と他の構成要素の一部とが重複していること、等でもよい。

[0020] また、本発明のデータ処理方法およびコンピュータプログラムには複数の手順を順番に記載してあるが、その記載の順番は複数の手順を実行する順番を限定するものではない。このため、本発明のデータ処理方法およびコンピュータプログラムを実施するときには、その複数の手順の順番は内容的に支障のない範囲で変更することができる。

[0021] さらに、本発明のデータ処理方法およびコンピュータプログラムの複数の手順は個々に相違するタイミングで実行されることに限定されない。このため、ある手順の実行中に他の手順が発生すること、ある手順の実行タイミングと他の手順の実行タイミングとの一部ないし全部が重複していること、等でもよい。

発明の効果

[0022] 本発明によれば、分散テーブルへのアクセス時のレイテンシの影響を削減し、処理の高速化を図る分散システム、情報処理装置のデータ処理方法およびプログラムが提供される。

図面の簡単な説明

[0023] 上述した目的、およびその他の目的、特徴および利点は、以下に述べる好適な実施形態、およびそれに付随する以下の図面によってさらに明らかになる。

[0024] [図1]本発明の実施の形態に係る分散システムの構成を示す機能ブロック図である。

[図2]本発明の実施の形態に係る分散システムのアクセス待ちリストの一例を示す図である。

[図3]本発明の実施の形態に係る分散システムのクライアント装置を実現する

コンピュータの構成を示すブロック図である。

[図4]本発明の実施の形態に係る分散システムにおけるクライアント装置の動作の一例を示すフローチャートである。

[図5]本発明の実施の形態に係る分散システムの利用形態の一つを示すブロック図である。

[図6]本発明の実施の形態に係る分散システムの利用形態の一つを示すブロック図である。

[図7]本発明の実施の形態に係る分散システムのクライアント装置の決定部の第1の方法による処理手順の一例を示すフローチャートである。

[図8]本発明の実施の形態に係る分散システムのクライアント装置の決定部の第2の方法による処理手順の一例を示すフローチャートである。

[図9]本発明の実施の形態に係る分散システムのクライアント装置の決定部の第3の方法による処理手順の一例を示すフローチャートである。

[図10]本発明の実施の形態に係る分散システムのクライアント装置の決定部の第4の方法による処理手順の一例を示すフローチャートである。

[図11]本発明の実施の形態に係る分散システムのクライアント装置の決定部の第4の方法による処理手順の他の例を示すフローチャートである。

[図12]本発明の実施の形態に係る分散システムのクライアント装置の決定部の第4の方法による処理手順の他の例を示すフローチャートである。

[図13]本発明の実施の形態に係る分散システムのクライアント装置の決定部の第5の方法による処理手順の一例を示すフローチャートである。

[図14]本発明の実施の形態に係る分散システムのクライアント装置の決定部の第6の方法による処理手順の一例を示すフローチャートである。

[図15]本発明の実施の形態に係る分散システムのクライアント装置の決定部の第7の方法による処理手順の一例を示すフローチャートである。

[図16]本発明の実施の形態に係る分散システムのクライアント装置の決定部の第7の方法による処理手順の他の例を示すフローチャートである。

[図17]本発明の実施の形態に係る分散システムの構成を示す機能ブロック図

である。

[図18]本発明の実施の形態に係る分散システムのアクセス待ちリストにライトアクセスが追加で登録された場合の例を示す図である。

[図19]本発明の実施の形態に係る分散システムのアクセス待ちリストにアキュムレートアクセスが追加で登録された場合の例を示す図である。

[図20]本発明の実施の形態に係る分散システムのアクセス待ちリストにアキュムレートアクセスの後、さらにリードアクセスが追加で登録された場合の例を示す図である。

[図21]本発明の実施の形態に係る分散システムのリード関数実行部の動作の一例を示すフローチャートである。

[図22]本発明の実施の形態に係る分散システムのライト関数実行部の動作の一例を示すフローチャートである。

[図23]本発明の実施の形態に係る分散システムのアキュムレート関数実行部の動作の一例を示すフローチャートである。

[図24]本発明の実施の形態に係る分散システムのリードアクセス発行部の動作の一例を示すフローチャートである。

[図25]MapReduceの動作を表す図である。

[図26]非特許文献1のシステムの構成を表す図である。

[図27]本発明の実施の形態に係る情報処理装置の論理的な構成を示す機能ブロック図である。

発明を実施するための形態

[0025] 以下、本発明の実施の形態について、図面を用いて説明する。尚、すべての図面において、同様な構成要素には同様の符号を付し、適宜説明を省略する。

[0026] (第1の実施の形態)

図1は、本発明の実施の形態に係る分散システム1の構成を示す機能ブロック図である。

本発明の実施の形態に係る分散システム1は、複数のサーバ（分散テーブ

ルサーバ10)がそれぞれ有する複数の分散テーブル(不図示)からデータを読み出すリード(read)アクセス処理のキー(Key)、および、キーに基づき読み出した結果(バリュー: Value)を用いて実行する処理を定めるコールバック(Callback)、を受け付ける受付部102と、受け付けたリードアクセス処理を待機させ、該当する分散テーブルサーバ10に後で発行する待機リードアクセスとしてキーとコールバックをアクセス待ちリスト104に登録する登録部106と、アクセス待ちリスト104に登録された待機リードアクセスについて、所定の条件に従い、待機リードアクセスを発行する発行先の分散テーブルサーバ10および発行タイミングを決定する決定部108と、決定部108が決定した発行先の分散テーブルサーバ10に、決定した発行タイミングで、待機リードアクセスを発行し、分散テーブルサーバ10から当該アクセス結果(バリュー)を受け取るリードアクセス発行部110と、リードアクセス発行部110が受け取ったアクセス結果を用いて、アクセス待ちリスト104に登録された該当する待機リードアクセスのコールバックを実行し、実行したコールバックの待機リードアクセスをアクセス待ちリスト104から削除するコールバック実行部112と、を備える。

[0027] 図1に示すように、本実施形態では、分散システム1は、複数の分散テーブルサーバ10(図1では、3つの分散テーブルサーバS1、S2、およびS3)とネットワーク3を介して接続されるクライアント装置100を備える。

[0028] 本発明の分散システム1は、複数のマシン上に分散して存在するテーブルに対して値を読み書きするシステムであり、特に、読み書きに用いるキーがランダムの場合でも高速にアクセスできる分散テーブルアクセスシステムに好適に適用できる。

[0029] 本発明の分散システム1は、たとえば、大規模なデータを対象とした機械学習やデータマイニング等の分析を行うものに好適に適用できる。本発明の分散システム1は、たとえば、Map Reduceのような大規模なデータを扱うシステムにおいて、データ処理部が、ランダムアクセスする共有デー

タが必要な場合に、そのアクセスを高速化するという用途に適用できる。
図5に、データ処理部90が実行するMapReduceプログラムのmap関数やreduce関数から分散テーブルを利用するシステムに、本発明のクライアント装置（図中、「分散テーブルクライアント92」と示す）を適用した例を示す。

[0030] また、本発明の分散システム1は、Webアプリケーションにおいて、Webサーバやアプリケーション（AP：Application）サーバがデータベースの内容をキャッシュするために分散テーブルを利用するという用途にも適用可能である。図6に、WebサーバまたはAPサーバ（図中、「Webサーバ/APサーバ94」と示す）がデータベース96の内容をキャッシュするために分散テーブルを利用する場合などに、本発明のクライアント装置（図中、「分散テーブルクライアント98」と示す）を適用した例を示す。

本発明のクライアント装置100は、図5または図6の分散テーブルクライアント（92または98）に相当する。

[0031] 図1に戻り、本実施形態のクライアント装置100は、分散テーブルサーバ10にネットワーク3を介して接続されるサーバコンピュータやパーソナルコンピュータ、またはそれらに相当する情報処理装置（図3のコンピュータ60）により実現することができる。また、クライアント装置100は、仮想サーバなどにより構成されてもよい。なお、図1では、クライアント装置100は1つのみ示されているが、本発明の分散システム1において、図示されない複数のクライアント装置100が分散テーブルサーバ10にアクセスすることができる。

[0032] 図1の本実施形態の分散システム1のクライアント装置100の各構成要素は、図3のCPU（Central Processing Unit）62、RAM（Random Access Memory）66、RAM66にロードされた図1の構成要素を実現するプログラム70、そのプログラム70を格納するROM（Read Only Memory）64、ネットワーク3接続用インタフェースを含むI/O（Input/Output）68を有する任意のコンピュータ60のハードウェアとソフトウェアの任意

の組合せによって実現される。CPU 62は、コンピュータ60の各要素とバス69を介して接続され、各要素とともにコンピュータ60全体を制御する。そして、その実現方法、装置にはいろいろな変形例があることは、当業者には理解されるところである。以下に説明する各機能ブロック図は、ハードウェア単位の構成ではなく、論理的な機能単位のブロックを示している。なお、コンピュータ60は、I/O 68を介して図示されない入出力装置と接続することもできる。

また、以下に示す各図において、本発明の本質に関わらない部分の構成については省略してあり、図示されていない。

[0033] 本実施形態において、分散システム1は、たとえば、データセンタ等に設置された複数の分散テーブルサーバ10を備えたクラスタまたはグリッドシステムであるとする。

図1において、クライアント装置100は、受付部102と、アクセス待ちリスト104と、登録部106と、決定部108と、リードアクセス発行部110と、コールバック実行部112と、を備える。

[0034] 受付部102は、複数の分散テーブルサーバ10がそれぞれ有する複数の分散テーブル（不図示）からデータを読み出すリードアクセス処理のキー、および、キーに基づき読み出した結果（バリュー）を用いて実行する処理を定めるコールバック、を受け付ける。

[0035] たとえば、分散テーブルへのアクセス処理は、一つのプログラムテキスト上の複数の箇所に現れる。

分散テーブル上のデータアクセスはレイテンシが大きく、性能劣化の原因となる。これを改善するためには、分散テーブルへのアクセス処理では、できるだけ複数のデータを一度に読み書きすればよいが、プログラム上、離れた別の場所に現れる分散テーブルへのアクセスを1つにまとめるのは容易ではない。ここで、プログラムとは、クライアント装置100が実行する分散テーブルへのアクセスを含むプログラム、あるいは、クライアント装置100を介して分散テーブルにアクセスする少なくとも1つの他のコンピュータ

で実行されるプログラムを含むことができる。

[0036] 本発明では、プログラム上、離れた別の場所に現れる複数の分散テーブルに対するアクセスを1つにまとめるため、コールバックと呼ばれる仕組みを用いる。コールバックとは、何らかの処理を行なった後に、続けて処理を行ないたい場合、この後続の処理をオブジェクトとして登録しておく仕組みである。

本発明では、これらのアクセス処理をまとめてアクセス先であるサーバ毎にまとめて行うことで、高速化を図る。

[0037] 本発明において、リードアクセス処理とは、たとえば、分散テーブルからデータを読み出す処理を行うリード (read) 関数等のAPI (Application Program Interface) によって実行される処理を含むことができる。本実施形態では、リード関数に、分散テーブルからデータを読み出すためのキー (Key) と、読み出した結果を用いて処理を実行するコールバック (Callback) を引数として指定する。受付部102は、プログラムを実行時にリード関数が呼び出されたとき、リード関数で引数として指定されているキーとコールバックを受け付ける。

[0038] アクセス待ちリスト104は、図2(a)に示すように、リードアクセス処理の対象となるデータを保持する分散テーブルを有する分散テーブルサーバ10 (担当サーバ) と、待機リードアクセスとしてキー (Key) とコールバック (Callback) のセットとを対応付けて保持する。

なお、アクセス待ちリスト104において、分散テーブルサーバ10 (担当サーバ) の情報として、予め付与された識別情報、またはネットワーク3上のIP (Internet Protocol) アドレス等を保持することができる。

[0039] 本実施形態のクライアント装置100において、キーを担当する分散テーブルサーバ10を探索する方法は、分散システムの汎用の手法を用いて行うことができ、特に限定されない。たとえば、分散テーブルサーバ10を探索する方法は、キーのハッシュ値を取って、サーバ数でmod (除算した結果出た余り) を取ることで、分散テーブルサーバ10を求めたり、予め準備さ

れた対応関係を用いて分散テーブルサーバ10を探索してもよい。

[0040] 本実施形態では、クライアント装置100は、担当するサーバ毎（図2（a））に、待機リードアクセスとしてキーとコールバックのセットを登録する構成としているが、これに限定されるものではない。たとえば、クライアント装置100は、キー毎（図2（b））、または予め属性等によりグループ分けされた複数のキー毎に、コールバックを保持する構成としてもよい。あるいは、クライアント装置100は、サーバ毎、かつ、キー毎（またはキーグループ毎）に分類してコールバックを保持する構成としてもよい。また、キー毎（またはキーグループ毎）にコールバックを保持する構成の場合、リードアクセス発行部110は、アクセスするキー（またはキーグループ）を担当する分散テーブルサーバ10を探して、待機リードアクセスを発行することができる。キー毎（またはキーグループ毎）に分類して保持している構成の場合、リードアクセス発行部110は、サーバの担当キー（またはキーグループ）の変更にも柔軟に対応できる。

[0041] 図1に戻り、登録部106は、受け付けたリードアクセス処理を直ぐに実行せずに、待機させ、該当する分散テーブルサーバ10（担当サーバ）に後で発行する待機リードアクセスとしてキーとコールバックをアクセス待ちリスト104に登録してバッファリングする。後述するように、本実施形態において、登録部106は、アクセス待ちリスト104に待機リードアクセスを登録したとき、登録したことを決定部108に通知してもよい。なお、登録部106は、リード関数を実行するリード関数実行部とすることができ、リード関数実行部がアクセス待ちリスト104に登録したことを通知してもよい。

[0042] 決定部108は、アクセス待ちリスト104に登録された待機リードアクセスについて、所定の条件に従い、待機リードアクセスを複数まとめて発行する発行先の分散テーブルサーバ10およびその発行タイミングを決定する。たとえば、決定部108は、アクセス待ちリストに登録された待機リードアクセスの件数が一定数以上になったときを、発行タイミングであると決定

する。

[0043] すなわち、決定部 108 は、分散テーブルサーバ 10 毎に、アクセス待ちリスト 104 に登録された待機リードアクセス処理の数が一定数以上になったときに、その分散テーブルサーバ 10 を発行先として、その分散テーブルサーバ 10 に対するリードアクセス処理の発行タイミングとする。あるいは、決定部 108 は、たとえば、キー毎に、アクセス待ちリスト 104 に登録された待機リードアクセス処理の数が一定数以上になったときに、そのキーを担当する分散テーブルサーバ 10 をリードアクセス処理の発行先として、その分散テーブルサーバ 10 に対するリードアクセス処理の発行タイミングを決定してもよい。

[0044] 以後、この待機処理の発行タイミングの決定方法を第 1 の方法と呼ぶものとする。この第 1 の方法では、アクセス待ちリスト 104 の保持に必要とするメモリ量や担当分散テーブルサーバ 10 毎の通信速度の違いは考慮されない。したがって、クライアント装置 100 では、第 1 の方法を、後述する他の方法と組み合わせた中から、分散システム 1 のシステム構成を考慮して、最適な方法を選択して採用することが望ましい。

[0045] また、リードアクセス処理の発行タイミングは、サーバ毎ではなく、キー毎に、または、予め属性等によりグループ分けされた複数のキー毎に、アクセス待ちリスト 104 に登録された待機リードアクセス処理の数が一定数以上になったとき、としてもよい。そして、その待機リードアクセス処理の数が一定数以上になったキーを担当する分散テーブルサーバ 10 をリードアクセス処理の発行先とする。そして、その複数の待機リードアクセスは、その分散テーブルサーバ 10、あるいは、その分散テーブルサーバ 10 の当該キー、または当該キーを含む同じグループの複数のキーに対して発行されてもよい。

[0046] リードアクセス発行部 110 は、決定部 108 が決定した発行先の分散テーブルサーバ 10 に、決定した発行タイミングで、待機リードアクセスを複数まとめて発行し、分散テーブルサーバ 10 から当該アクセス結果（バリュ

一)を受け取る。すなわち、同一サーバのリードアクセスを複数まとめて発行する。

[0047] リードアクセスをまとめて発行せず、その度、実行する場合、1回の処理にかかる時間は、レイテンシ+リードデータサイズ/スループットとなる。この処理時間が、リードアクセス毎にかかることになる。一方、リードアクセスを複数まとめて発行すると、処理時間のうち、レイテンシは1処理分だけで済むこととなり、処理時間を短縮できる。

[0048] コールバック実行部112は、リードアクセス発行部110が受け取ったアクセス結果（バリュー）を用いて、アクセス待ちリスト104に登録された該当する待機リードアクセスのコールバックを実行し、実行したコールバックの待機リードアクセスをアクセス待ちリスト104から削除する。

[0049] 本実施の形態のクライアント装置100では、コンピュータプログラム70（図3）に対応する各種の処理動作をコンピュータ60のCPU62（図3）が実行することにより、前述のような各種ユニットが各種機能として実現される。

本実施形態のコンピュータプログラム70は、情報処理装置（クライアント装置100）を実現させるためのコンピュータ60に、複数の分散テーブルサーバ10がそれぞれ有する複数の分散テーブルからデータを読み出すリードアクセス処理のキー、および、キーに基づき読み出した結果を用いて実行する処理を定めるコールバック、を受け付ける手順、受け付けたリードアクセス処理を待機させ、該当する分散テーブルサーバ10に後で発行する待機リードアクセスとしてキーとコールバックをアクセス待ちリスト104に登録する手順、アクセス待ちリスト104に登録された待機リードアクセスについて、所定の条件に従い、待機リードアクセスを発行する分散テーブルサーバ10および発行タイミングを決定する手順、決定した分散テーブルサーバ10に、決定した発行タイミングで、待機リードアクセスを発行し、分散テーブルサーバ10から当該アクセス結果を受け取る手順、受け取ったアクセス結果を用いて、アクセス待ちリスト104に登録された該当する待機

リードアクセスのコールバックを実行し、実行したコールバックの待機リードアクセスをアクセス待ちリスト104から削除する手順、を実行させるように記述されている。

[0050] 本実施形態のコンピュータプログラム70は、コンピュータ60で読み取り可能な記録媒体に記録されてもよい。記録媒体は特に限定されず、様々な形態のものが考えられる。また、プログラム70は、記録媒体からコンピュータ60のメモリ（ROM64またはRAM66）にロードされてもよいし、ネットワークを通じてコンピュータ60にダウンロードされ、メモリにロードされてもよい。

[0051] 上述のような構成において、本実施の形態の分散システム1におけるクライアント装置100によるデータ処理方法を以下に説明する。図4は、本実施形態の分散システム1の動作の一例を示すフローチャートである。以下、図1および図4を用いて説明する。

本発明の実施の形態に係る情報処理装置（図1のクライアント装置100）のデータ処理方法は、クライアント装置100が、複数の分散テーブルサーバ10（図1）がそれぞれ有する複数の分散テーブルからデータを読み出すリードアクセス処理のキー、および、キーに基づき読み出した結果を用いて実行する処理を定めるコールバック、を受け付け（ステップS101）、受け付けたリードアクセス処理を待機させ、該当する分散テーブルサーバ10に後で発行する待機リードアクセスとしてキーとコールバックをアクセス待ちリスト104に登録し（ステップS103）、アクセス待ちリスト104に登録されたリードアクセスについて、所定の条件に従い、待機リードアクセスを発行する分散テーブルサーバ10および発行タイミングを決定し（ステップS105）、決定した分散テーブルサーバ10に、決定した発行タイミングで（ステップS107のYES）、待機リードアクセスを発行し、分散テーブルサーバ10から当該アクセス結果を受け取り（ステップS109）、受け取ったアクセス結果を用いて、アクセス待ちリスト104に登録された該当する待機リードアクセスのコールバックを実行し（ステップS1

11)、実行したコールバックの待機リードアクセスをアクセス待ちリスト104から削除する(ステップS113)。

[0052] 図7は、本実施形態の分散システム1のクライアント装置100の決定部108のリードアクセスの発行タイミングと発行先サーバの決定処理の手順の例を示すフローチャートである。本実施形態では、決定部108は、上述した第1の方法により決定処理を行う。

まず、決定部108が、登録部106(リード関数)からアクセス待ちリスト104に登録したとの通知を受け取る度に(ステップS121のYES)、本処理を開始する。

[0053] そして、決定部108が、アクセス待ちリスト104を参照し、待機リードアクセス処理の数が一定数以上の分散テーブルサーバ10があるか否かを判定する(ステップS123)。一定数以上のサーバがない場合は(ステップS123のNO)、手順は、ステップS121に戻り、決定部108は、次の通知の到来まで待機する(ステップS121のNO)。

一定数以上のサーバがある場合(ステップS123のYES)、決定部108が、当該サーバに対する待機していた複数のリードアクセスをまとめて発行する(ステップS125)。

[0054] あるいは、登録部106が、アクセス待ちリスト104に登録を行う度に、分散テーブルサーバ10毎に登録数をカウントし、メモリ(RAM66等)に記憶してもよい。そして、決定部108は、メモリを参照し、各カウント値が一定値以上になった分散テーブルサーバ10があるか否かを監視してもよい。

[0055] 以上説明したように、本発明の実施の形態に係る分散システム1によれば、クライアント装置100が、任意のプログラムを対象として、プログラム上の任意の場所に現れる分散テーブルサーバ10へのリード関数による複数のリードアクセス処理を、その都度リードアクセス毎に行うのではなく、アクセス待ちリスト104に登録して一定数以上待機させることで、サーバ毎にまとめて実行することができる。そして、この構成により、本発明の実施

の形態に係る分散システム１において、プログラム上に複数現れるリードアクセス処理を、その都度処理する場合に比較して、リードアクセス時にかかるレイテンシによるアクセス処理速度への影響を削減することができ、高速化を図ることができる。

[0056] 特に、本発明の実施の形態に係る分散システム１によれば、クライアント装置１００は、プログラム上の何処にいつ現れるか分からないような複数のリードアクセス処理に好適に適用することが可能になる。その理由は、クライアント装置１００が、リードアクセス処理が現れた時に、その処理を受け付けて、適切に待機させる構成を有しているからである。

[0057] (第２の実施の形態)

本実施形態の分散システム１は、上記実施形態とは、待機アクセス処理の発行タイミングの決定方法が異なり、システム構成を考慮し、メモリ量やクラスタシステムのノード（分散テーブルサーバ１０）間のネットワーク的な距離を考慮して、高速化に最適なノードを選択して待機アクセス処理を実行する点で相違する。

本実施形態の分散システム１の構成は、図１の上記実施形態の分散システム１と同じであり、以下、図１を用いて説明する。

[0058] 上記実施形態では、クライアント装置１００は、分散テーブルへの複数のアクセスを１つにまとめて処理することで、レイテンシの影響を抑えることができ、処理の高速化を図ることができた。本実施形態では、クライアント装置１００は、さらに、この複数のアクセスを、どのタイミングでどのサーバに対するものを優先してアクセスするか、メモリ量やマシン間の通信速度の違い等、分散システム１のシステム構成等の条件を考慮して、決定することで、より性能の向上を図るものである。

[0059] はじめに、第２の方法として、メモリ量を考慮して発行タイミングと発行先サーバを決定する方法について説明する。

本実施形態において、決定部１０８は、アクセス待ちリスト１０４のメモリ量が一定量以上になった場合、発行タイミングになったと判断し、メモリ

量が一定量以上になった時点で、アクセス待ちリスト 104 に登録された待機リードアクセスの件数が多い分散テーブルサーバ 10 を待機リードアクセスの発行先として選択する。

[0060] アクセス待ちリスト 104 のメモリ量は、キーとコールバックが必要とするメモリ量を加算していくことで求めることができる。ただし、複数の Callback オブジェクト間で共有されるメモリがある場合は、コールバックが必要とするメモリ量を正確に求めることが難しい可能性もある。その場合は、リード関数の引数に、アクセス待ちリスト 104 の登録に必要となるメモリ量を、プログラマが指定するようにしてもよい。アクセス待ちリスト 104 の利用メモリ量は、コールバックの実行が完了して当該オブジェクトで利用しているメモリが解放され次第、減算する。

なお、アクセス待ちリスト 104 がキー毎に分類されて登録されている場合、アクセス待ちリスト 104 に登録された待機リードアクセスの件数が多いキーに対応する分散テーブルサーバ 10 を待機リードアクセスの発行先として選択する。

なお、この第 2 の方法では、担当サーバごとの通信速度の違いは考慮されない。したがって、クライアント装置 100 では、第 2 の方法を、他の方法と組み合わせた中から、分散システム 1 のシステム構成を考慮して、最適な方法を選択して採用することが望ましい。

[0061] 次に、第 3 の方法として、マシン間の通信速度の違いを考慮して発行タイミングと発行先サーバを決定する方法について説明する。

本実施形態において、決定部 108 は、各分散テーブルサーバ 10 の性能情報に基づいて求まる、キー 1 つあたりにかかるアクセス結果（バリュー）の取得時間に基づいて、待機リードアクセスの発行先となるサーバを選択する。

第 3 の方法では、決定部 108 は、アクセス待ちリスト 104 のメモリ量が一定量以上になった場合、1 つのキーあたりにかかるバリュー取得時間が最も少ない分散テーブルサーバ 10 を待機リードアクセスの発行先として選

択する。すなわち、決定部 108 は、通信速度が速い分散テーブルサーバ 10 を発行先として選択する。

[0062] 具体的には、1つのキーあたりにかかるバリュ取得時間は、分散テーブルサーバ 10 ごとに、下記の式(1)で求めることができる。

[数1]

$$\{\text{通信レイテンシ} + (\text{複数のキーに対応するバリュの総バイト数}) / \text{スループット}\} / \text{キー数} \quad \cdots \text{式(1)}$$

[0063] ここで、本実施形態の分散システム 1 がクラスタシステムであることを考慮すると、ノード（分散テーブルサーバ 10）間で、互いのネットワーク的な距離が近いノードと遠いノードが存在する。ここで、ネットワーク的な距離は、キー 1 つあたりにかかるアクセス結果（バリュ）の取得時間で評価することができ、各ノードから他のノードへの通信レイテンシおよびスループット値から求めることができる。

[0064] キー 1 つあたりにかかるアクセス結果（バリュ）の取得時間は、ノード間の距離が同じならば、待機アクセス処理が最も多くたまったノードが、最小となる。

また、待機アクセス処理数が同じならば、ノード間の距離が最も近いノードが、キー 1 つあたりにかかるアクセス結果（バリュ）の取得時間が最小となる。

[0065] よって、クライアント装置 100 では、ノード間の距離が同じならば、待機アクセス処理が最も多くたまったノードから、待機アクセス処理を実行し、待機アクセス処理数が同じならば、距離が近いノードから、待機アクセス処理を実行するのが望ましい。すなわち、距離が遠いノードについては、より多くの待機アクセス処理をためてから実行するのが望ましい。本実施形態では、決定部 108 が、1つのキーあたりにかかるバリュ取得時間が最も少ないノードを、待機リードアクセスの発行先として選択することで、最もレイテンシの影響が小さくなるノードを選択できる。

[0066] ここで、クライアント装置 100 は、担当サーバごとの通信速度の違いを考慮するために必要な、各担当分散テーブルサーバ 10 への通信レイテンシ

およびスループット等の情報を、予め得ておく。これらの情報は、測定した結果から得た値やシステムの構成（スイッチ／ルータの段数を含むネットワーク構成等）から算出した値をクライアント装置１００にあらかじめ登録しておいてもよい。あるいは、クライアント装置１００が、システムの初期化時に必要なデータを測定して、これらの情報を求める算出部（不図示）を備えるようにしてもよい。

[0067] なお、アクセス待ちリスト１０４には、同一キーに対するリードアクセスが複数登録される可能性があるが、決定部１０８は、式（１）において、それらを１つにまとめて１つのキーとしてカウントして、キー数を求める。バリューのバイトサイズがキーによって異なるなど、複数のキーに対応するバリューの総バイト数を求めるのが難しい場合は、決定部１０８は、式（１）において、近似値を用いればよい。また、通信レイテンシの影響の方がスループットの影響よりも圧倒的に大きな場合は、決定部１０８は、式（１）において、（複数のキーと対応するバリューの総バイト数）／スループットの項を削除して近似してもよい。

[0068] このようにすることで、全ての分散テーブルサーバ１０が同じ条件であるならば、クライアント装置１００は、第３の方法においても、第２の方法と同様に、最もアクセス待ち数が多いサーバに対してリードアクセスを発行することができる。もし、全ての分散テーブルサーバ１０でアクセス待ち数が同じであれば、ネットワーク的に近いサーバが選ばれる。すなわち、ネットワーク的に遠いサーバについては、クライアント装置１００は、なるべく多くのリードアクセスをためてから発行することになる。

[0069] 次に、決定部１０８が発行タイミングと発行先サーバを決定する第４の方法について説明する。

第４の方法では、決定部１０８は、第３の方法で説明した各分散テーブルサーバ１０の「１つのキーあたりにかかるバリュー取得時間」を所定のタイミングで算出し、全分散テーブルサーバ１０の中での最小値を求め、最小値に基づいて発行タイミングと発行先サーバを決定する。

[0070] 決定部108が、「1つのキーあたりにかかるバリュ取得時間」を算出するタイミングは、リードアクセスがアクセス待ちリスト104に登録される度、リードアクセスがアクセス待ちリスト104に所定回数登録される度、または、一定時間毎である。決定部108は、算出された最小値が閾値以下になった時を発行タイミングと判断し、最小値となったサーバを発行先とする。ここで、決定部108は、アクセス待ちリスト104への登録の有無や登録回数を、登録部106からの通知に基づいて検出することができる。

決定部108は、「1つのキーあたりにかかるバリュ取得時間」を、上記の第3の方法と同様に、サーバごとに上述した式(1)で求めることができ、詳細な説明は省略する。

[0071] ここで、決定部108は、リードアクセスがアクセス待ちリスト104に登録されたことを、登録部106(リード関数)からの登録通知により検知できる。この通知(リードアクセス)が所定回数以上来るまで待つ理由は、1つのキーあたりにかかるバリュ取得時間の最小値が閾値以下のサーバがあるかどうかを、リード関数からアクセス待ちリスト104に登録した通知を受け取るたびに計算するのが非効率な場合に、まとめて計算してそのオーバーヘッドを削減するためである。

[0072] この第4の方法では、クライアント装置100は、アクセス待ちリスト104のメモリ量を参照しないため、設定された閾値の大小によって、過小あるいは過大にアクセス待ちリスト104用のメモリを消費する可能性がある。したがって、クライアント装置100では、第4の方法を、他の方法と組み合わせた中から、分散システム1のシステム構成を考慮して、最適な方法を選択して採用することが望ましい。

[0073] また、第5の方法として、以下の方法も考えられる。

決定部108は、ある一定時間(たとえば、10ms)リードアクセスが発行されていない分散テーブルサーバ10をリードアクセスの発行先として選択する。発行タイミングは、一定時間リードアクセスが発行されていないサーバが存在したときとなる。

[0074] この方法で一定時間リードアクセスが発行されていない分散テーブルサーバ10を発行先とする理由は、図6に示したWebサーバ/APサーバ94に本発明の分散システム1を用いるような場合、長時間ユーザを待たせるのは望ましくないためである。ただし、この場合はアクセス待ちリスト104のメモリ量や、1つのキーあたりにかかるバリュ取得時間を参照しないため、この第5の方法は、必ずしも効率のよいサーバの選択にはならない。したがって、クライアント装置100では、第5の方法を、他の方法と組み合わせた中から、分散システム1のシステム構成を考慮して、最適な方法を選択して採用することが望ましい。

[0075] さらに、第6の方法では、プログラマがプログラム上で明示的にリードアクセスを発行するサーバおよび発行タイミングを指定する。たとえば、C++で記述した関数の一例では、flush（サーバ名）のような関数を実行することで、この関数が実行されたタイミングで、リードアクセス発行部110が、当該サーバへのリードアクセスを発行する。

[0076] また、決定部108は、第7の方法として、第1～第6の方法を少なくとも2つ組み合わせて決定方法として採用してもよい。詳細については、後述する。

[0077] 図8～図16は、本実施形態の分散システム1のクライアント装置100の決定部108の各方法による処理手順の例をそれぞれ示すフローチャートである。

図8の処理手順は、決定部108が上述した第2の方法を用いて決定処理を行う場合の例である。つまり、この処理手順は、メモリ量が一定以上の場合、待機リードアクセスの件数が最も多いサーバにリードアクセスを発行する決定処理の例である。

[0078] 図8の例では、決定部108が、リード関数からアクセス待ちリスト104に登録したとの通知を受け取る度に（ステップS121のYES）、本処理が開始する。通知を受け取り（ステップS121のYES）、かつ、アクセス待ちリスト104のメモリ量が一定量以上になった時（ステップS13

１のＹＥＳ）、決定部１０８が、発行タイミングになったと判断する。

[0079] そして、決定部１０８が、その時点で、アクセス待ちリスト１０４に登録された待機リードアクセスの件数が最も多い分散テーブルサーバ１０を発行先に決定する。そして、リードアクセス発行部１１０が、発行先に決定された分散テーブルサーバ１０に、リードアクセスを発行する（ステップＳ１３３）。

一方、メモリ量が一定量以上でない場合（ステップＳ１３１のＮＯ）、手順は、ステップＳ１２１に戻り、決定部１０８は、次の通知の到来まで待機する（ステップＳ１２１のＮＯ）。

[0080] 次に、図９の処理手順は、決定部１０８が上述した第３の方法を用いて決定処理を行う場合の例である。つまり、この処理手順は、メモリ量が一定以上の場合、１つのキーあたりにかかるバリュウ取得時間が最も少ないサーバにリードアクセスを発行する決定処理の例である。

図９の例では、リード関数からアクセス待ちリスト１０４に登録したとの通知を受け取り（ステップＳ１２１のＹＥＳ）、かつ、アクセス待ちリスト１０４のメモリ量が一定量以上になった時（ステップＳ１３１のＹＥＳ）、決定部１０８が、発行タイミングになったと判断する。ここまでの図９の処理手順は、図８の例と同じである。

[0081] そして、決定部１０８が、１つのキーあたりにかかるバリュウ取得時間が最も少ない分散テーブルサーバ１０を発行先に決定し、リードアクセス発行部１１０が、決定された発行先の分散テーブルサーバ１０に、リードアクセスを発行する（ステップＳ１４１）。

[0082] 次に、図１０～図１２の処理手順は、決定部１０８が上述した第４の方法を用いて決定処理を行う場合の例である。各図の手順は、１つのキーあたりにかかるバリュウ取得時間を算出（または、バリュウ取得時間の最小値と閾値を比較）するタイミングがそれぞれ異なる。

[0083] 図１０は、アクセス待ちリスト１０４にリードアクセスが登録される度に、第４の方法による決定処理が行われる例を示す。図１１は、アクセス待ち

リスト１０４にリードアクセスが所定回数以上登録された場合に、第４の方法による決定処理が行われる例を示す。図１２は、所定の時間間隔で第４の方法による決定処理が行われる例を示す。

まず、図１０の例では、決定部１０８が、リード関数からアクセス待ちリスト１０４に登録したとの通知を受け取る度に（ステップＳ１２１のＹＥＳ）、本処理を開始する。

[0084] そして、決定部１０８が、１つのキーあたりにかかるバリュウ取得時間の最小値が閾値以下のサーバがあるかを判別する（ステップＳ１５１）。最小値が閾値以下のサーバがない場合は（ステップＳ１５１のＮＯ）、手順は、ステップＳ１２１に戻り、決定部１０８は、次の通知の到来まで待機する（ステップＳ１２１のＮＯ）。

最小値が閾値以下のサーバがある場合（ステップＳ１５１のＹＥＳ）、決定部１０８がそのサーバを発行先と決定し、リードアクセス発行部１１０が、当該サーバに対するリードアクセスを発行する（ステップＳ１２５）。

[0085] また、図１１の例では、決定部１０８が、リード関数からアクセス待ちリスト１０４に登録したとの通知を受け取り（ステップＳ１２１のＹＥＳ）、かつ、決定部１０８が、通知を受け取った回数をカウントしておき、所定回数以上、通知を受け取った場合に（ステップＳ１６１のＹＥＳ）、本処理を開始する。以降の手順は図１０と同様である。なお、決定部１０８が本処理を実行した時、カウントはリセットされる。すなわち、所定回数毎に、本処理が実行されることとなる。

[0086] また、図１２の例では、図示されないタイマにより、所定の時間間隔を計測し、決定部１０８がタイマからの通知（タイマ割り込み）を受け取る度に（ステップＳ１７１のＹＥＳ）、すなわち、一定時間毎に、本処理を開始する。以降の手順は図１０と同様である。なお、タイマの時間は、たとえば、１０ｍｓとすることができ、予め設定された値であってもよいし、条件に応じて変更されてもよい。

また、クライアント装置１００において、上記図１０または図１１と、図

12との手順をそれぞれ組み合わせてもよい。

[0087] さらに、図13の処理手順は、決定部108が上述した第5の方法を用いて決定処理を行う場合の例である。図13の例では、図示されないタイマにより、所定の時間間隔を計測し、決定部108がタイマからの通知（タイマ割り込み）を受け取る度に（ステップS171のYES）、すなわち、一定時間毎に、本処理が開始する。そして、決定部108が、一定時間以上、リードアクセスが発行されていない分散テーブルサーバ10があるか否かを判定する（ステップS181）。リードアクセスが発行されていない分散テーブルサーバ10がない場合は（ステップS181のNO）、手順は、ステップS171に戻り、決定部108は、次の通知の到来まで待機する（ステップS171のNO）。リードアクセスが発行されていない分散テーブルサーバ10がある場合（ステップS181のYES）、決定部108が、そのサーバを発行先に決定し、リードアクセス発行部110が、当該サーバに対するリードアクセスを発行する（ステップS125）。

[0088] また、図14の処理手順は、決定部108が上述した第6の方法を用いて決定処理を行う場合の例である。図14の例では、まず、決定部108が、リードアクセスを発行するサーバおよび発行タイミングを指定した関数を呼び出す（ステップS191）。関数で指定されたタイミングになった時（ステップS193のYES）、決定部108が、指定された分散テーブルサーバ10を発行先に決定し、発行先の分散テーブルサーバ10に対して、リードアクセス発行部110がリードアクセスを発行する（ステップS195）。

[0089] また、第7の方法として、決定部108が第1～第6の方法を少なくとも2つ組み合わせて決定方法として採用した場合の処理手順を図15と図16に示す。ここで、図15と図16、さらに図14のフローは、後述するように並列的に実行することができる。

第4の方法のように「1つのキーあたりにかかるバリュ取得時間」が閾値以下のサーバがあれば（図15のステップS151のYES）、リードア

クセス発行部 110 が、当該サーバに対してリードアクセスを発行する（図 15 のステップ S 125）。

[0090] このようなサーバが無くても（図 15 のステップ S 151 の NO）、第 3 の方法のように、アクセス待ちリスト 104 のメモリ量が一定量以上になった場合（図 15 のステップ S 131 の YES）、決定部 108 が「1つのキーあたりにかかるバリュウ取得時間」が最も少ないサーバを発行先を選び、リードアクセス発行部 110 がリードアクセスを発行する（図 15 のステップ S 141）。これらの条件が成立しなくても（図 15 のステップ S 131 の NO、かつステップ S 151 の NO）、第 5 の方法のようにある一定時間リードアクセスが発行されていないサーバに対して（図 16 のステップ S 181 の YES）、リードアクセス発行部 110 がリードアクセスを発行する（図 16 のステップ S 125）。また、第 6 の方法のようにプログラム上で明示的に指定された場合もリードアクセスを発行する（図 14 のステップ S 195）。

[0091] また、これらの方法でリードアクセスを発行する場合、リードアクセス発行部 110 は、この発行をリードアクセス以外の処理や以前発行したリードアクセスと同時並行に、オーバーラップして実行してよい。また、第 1 から第 4、および第 6、第 7 の方法でリードアクセスを発行する場合、リードアクセス発行部 110 によるリードアクセスの発行の実施は、リード関数が実行されることが主な契機になる。この契機となるリード関数の中からリードアクセスの発行を行い、リード関数が終了した時点でリードアクセスの発行が終了しているように実装すれば、オーバーラップは行われない。さらに、契機となるリード関数の中からリードアクセスの発行を行う際、別スレッドでリードアクセスの発行を行えば、当該リードアクセスの発行を、リードアクセス以外の処理や他のリードアクセスの発行とオーバーラップして実行できる。

[0092] リードアクセス発行部 110 が、第 5 の方法や、第 7 の方法で「一定時間」リードアクセスが発行されていないサーバに対して（図 16 のステップ S 181 の YES）、リードアクセスを発行する場合、あるいは、第 4 の方法

で「一定時間」ごとに「1つのキーあたりにかかるバリュウ取得時間」を計算して、閾値を下回るサーバに対して（図15または図16のステップS151のYES）、リードアクセスを発行する場合、これらの「一定時間」ごとに行う判定をそれぞれ独立したスレッドで行い、当該スレッドでリードアクセスの発行をそれぞれ行うことで、オーバーラップを実現することができる。このように、クライアント装置100では、非同期にオーバーラップしてリードアクセスを実行させることで処理時間を有効活用することができる。

[0093] 以上説明したように、本発明の実施の形態に係る分散システム1によれば、上記実施形態と同様な効果を奏するとともに、メモリ量やマシン間の通信速度の違い等を考慮に入れた最適なアクセス方法を提供できる。

その理由は、決定部108が、アクセス待ちリスト104で消費されているメモリが一定量を越えると、1要素当たりの取得時間が最小のサーバを選択して当該サーバに対して複数のキーに対するリードアクセスを発行することで、マシンのメモリをできるだけ活用しながら、その時点で最も適切なサーバを選択できるためである。

[0094] 上記非特許文献1記載の技術では、分散テーブルに対する複数のアクセスをまとめることができたとしても、どのタイミングでどのサーバに対するものを優先して行なうのが最適であるかは明らかではなかった。

すなわち、上記非特許文献1記載の技術では、複数のアクセスをまとめるためには、なんらかの形で処理を遅延させる必要がある。そのために必要となるメモリ量を一定にした上で最も効率のよいアクセス方法を実現する必要がある。

[0095] さらに、実際にシステムが動作する環境を考えると、分散テーブルを構成する各マシンは必ずしも同じ条件ではない。すなわち、マシンの数が増えると、その接続は多段のツリーとして構成されることが多く、たとえば、同じラック内のマシンの方が別のラック内のマシンよりも通信速度が速いということが起こりうる。

[0096] 本発明の実施の形態に係る分散システム1によれば、このような環境を考

慮に入れた、最適なアクセス方法を実現することができる。

[0097] (第3の実施の形態)

図17は、本発明の実施の形態に係る分散システム1の構成を示す機能ブロック図である。

本実施形態の分散システム1は、上記実施形態とは、分散テーブルへのリードアクセスだけでなく、ライトアクセス、またはアキュムレートアクセスもアクセス待ちリスト104に登録する点で相違する。

[0098] 本発明の実施の形態に係る分散システム1は、複数の分散テーブルサーバ10にネットワーク3(図17には図示していない)を介して接続されるクライアント装置200を備える。なお、本実施形態のクライアント装置(図中、「分散テーブルクライアント」と示す)200は、上記第1の実施の形態または第2の実施の形態のクライアント装置100の決定部108と同様な決定処理手順を含む手順も実行することができる。

[0099] 本実施形態の分散システム1のクライアント装置200は、図1の上記実施形態のクライアント装置100と同じ、アクセス待ちリスト104と、リードアクセス発行部110と、コールバック実行部112と、を備える。さらに、クライアント装置200は、リード(read)関数実行部202と、ライト(write)関数実行部204と、アキュムレート(accumulate)関数実行部206と、ライト／アキュムレート(write/accumulate)アクセス発行部208と、決定部210と、を備える。

[0100] リード関数実行部202は、プログラムが実行されてリード関数が呼び出されたとき、リード関数のキー(Key)とコールバック(Callback)を、分散テーブル上でそのキーを担当する分散テーブルサーバ10ごとに分類して、アクセス待ちリスト104に登録する。すなわち、リード関数実行部202は、待機リードアクセスとしてキーとコールバックをアクセス待ちリスト104に登録してバッファリングする。本実施形態において、リード関数実行部202と、ライト関数実行部204と、アキュムレート関数実行部206は、上記実施形態の受付部102と登録部106に対応する。

- [0101] ライト関数実行部204は、プログラムが実行されてライト関数が呼び出されたとき、ライトアクセスの発行をライト／アキュムレートアクセス発行部208に要求、または、ライトアクセス処理を行う。ライト関数は、分散テーブルのデータに書き込みを行う関数であり、データを書き込むデータのキー（Key）と、書き込むデータ（Value）が引数として指定される。ライトアクセス処理とは、ライト関数で指定された引数のKeyのデータに、引数のValueのデータを書き込む処理である。
- [0102] アキュムレート関数実行部206は、プログラムが実行されてアキュムレート関数が呼び出されたとき、アキュムレートアクセスの発行をライト／アキュムレートアクセス発行部208に要求、または、アキュムレートアクセス処理を行う。アキュムレート関数は、引数として指定されたキー（Key）に対応する分散テーブルに元々あった値（データ）と、引数として指定された値（Value）の間で、引数として指定された演算処理を行い、その結果を、Keyのデータ（値）として格納するアクセスである。ここで、指定される演算処理は、結合法則が成り立つものであるとする。加算や乗算、最大値または最小値を求めるものが、演算処理の例となる。アキュムレートアクセス処理とは、アキュムレート関数で指定された引数のKeyのデータと引数で指定されたValueとの間で、引数で指定された演算処理を行い、引数のKeyのデータに演算結果の値を書き込む処理である。
- [0103] ライト／アキュムレートアクセス発行部208は、ライト関数実行部204またはアキュムレート関数実行部206からの要求または、決定部210によって決定された発行タイミングで、ライトまたはアキュムレートアクセスを、該当する分散テーブルサーバ10に発行する。
- [0104] 本実施形態の分散システム1のクライアント装置200において、受付部（ライト関数実行部204）は、分散テーブルのキーに対して値（バリュー）を書き込むライトアクセス処理をさらに受け付ける。そして、リードアクセス発行部110は、アクセス待ちリスト104を参照し、ライトアクセス処理と同じキーに対する待機リードアクセスが存在する場合、ライトアクセ

ス処理の値（バリュー）で待機リードアクセスのコールバックを実行し、アクセス待ちリスト１０４から当該待機リードアクセスを削除する。

[0105] また、受付部（ライト関数実行部２０４）は、リードアクセス処理より後に受け付けるライトアクセス処理を、先に処理することを禁止する禁止フラグをリードアクセス処理のキーとコールバックとともに受け付ける。そして、登録部（ライト関数実行部２０４）は、ライト関数実行部２０４が受け付けた禁止フラグを待機リードアクセスのキーおよびコールバックとともにアクセス待ちリスト１０４に登録する。さらに、登録部（ライト関数実行部２０４）は、アクセス待ちリスト１０４を参照し、ライトアクセス処理と同じキーに対する待機リードアクセスの禁止フラグが登録されている場合、ライトアクセス処理を待機させ、該当するサーバに後で発行する待機ライトアクセスとしてキーと値（バリュー）を前記アクセス待ちリスト１０４に登録する。

[0106] ここで、ライト関数またはアキュムレート関数が、対象とするキーのリードアクセスがアクセス待ちリスト１０４に登録されていない、または登録されていても追い越して処理してよい場合は、ライト関数実行部２０４またはアキュムレート関数実行部２０６は、即時ライト／アキュムレートアクセス発行部２０８にライトまたはアキュムレートアクセス発行を要求する。追い越せないリードアクセスがアクセス待ちリスト１０４に登録されている場合は、ライト関数実行部２０４またはアキュムレート関数実行部２０６は、ライトまたはアキュムレートアクセス処理を、待機アクセスとして、アクセス待ちリスト１０４に登録する。

[0107] 本実施形態において、決定部２１０は、アクセス待ちリスト１０４に登録されているリードアクセスと、それを待ち合わせているライトアクセス、アキュムレートアクセスについて、いつ（発行タイミング）、どの分散テーブルサーバ１０（発行先）に発行するかを決定する。

[0108] 本実施形態では、アクセス待ちリスト１０４で消費されているメモリが一定量を越えると、決定部２１０が、１要素当たりの取得時間が最小の分散テ

ーブルサーバ10を選択して、リードアクセス発行部110が、当該サーバに対して複数のキーに対するリードアクセスを発行する。そして、コールバック実行部112が、取得したキー（Key）とバリュー（Value）を用いて登録されていたコールバック（Callback）を実行する。その後、ライト／アキュムレートアクセス発行部208が、待ち合わせていたライトアクセス、またはアキュムレートアクセスを該当する分散テーブルサーバ10に発行する。

[0109] 決定部210における、いつ、どのサーバにリードを発行するか決定には、上記実施形態同様に、他の方法も考えられる。すなわち、決定方法には、一定時間以上リードアクセスが発行されていないサーバについて発行する方法、1要素当たりの取得時間が一定値以下のサーバについて発行する方法、プログラマが明示的に指定する方法、また、これらを組み合わせる方法がある。

[0110] 本実施形態のクライアント装置200は、このような構成、および動作を採用することで、プログラム上任意の場所に現れる複数のリードアクセスを一つにまとめて発行し、リードアクセスのレイテンシによる影響を削減することができる。また、本実施形態では、決定部210により、メモリ量やマシン間の通信速度の違い等を考慮に入れた最適なアクセス方法を提供できる。これにより、本発明の目的を達成することができる。

[0111] 以下、各アクセス処理について具体例を示しながら説明する。

まず、分散テーブルからデータを読む場合（リードアクセス）について説明する。

本発明では、分散テーブルサーバ10に対してアクセスする処理を含むプログラムにおいて、プログラム上、離れた場所に現れる複数の分散テーブルに対するアクセスを1つにまとめるため、コールバック（Callback）と呼ばれる仕組みを用いる。

ここでは、分散テーブルに対するアクセスを行なった後に実行する処理を、コールバックで実現する。

コールバック (Callback) を C++ で記述した一例を下記に示す。

```
class Callback { /* システムが提供 */
public:
    virtual void run (string Key, string Value) = 0;
};

class MyCallback : public Callback { /* 個々のプログラムで記述 */
public:
    MyCallback (...) {
        /* オブジェクト生成時にrun () を実行するために必要なデータを初期
        化 */
    }

    virtual void run (string Key, string Value) {
        /* Keyに対するValueを取得した結果、行う処理を記述 */
    }

private:
    /* run () を実行するために必要なデータを個々のプログラム毎に保存 */
};
```

[0112] ここでは、システムがCallbackクラスを提供し、それを継承することで個々のプログラムで利用するコールバック（ここではMyCallback）クラスを生成する例を示した。システムが提供するCallbackクラスは、run () メンバ関数がvirtual関数として提供されているだけの抽象クラスであるとする。

本システム上で実行され、分散テーブルサーバ10にアクセスする個々のプログラムは予め決まっていて、予め上記のMyCallbackクラスの宣言と下記read関数等の記述を個々のプログラムに記述することで、本システムは実現される。

[0113] また、分散テーブルに対するリードアクセスを行う際のプログラムインタフェース（read関数）をC++で記述した一例を下記に示す。

```
read (Key, new MyCallback (...)) ;
```

個々のプログラムにおいて、分散テーブルにリードアクセスする時、上記プログラムインタフェースのリード (read) 関数を呼び出せばよい。

[0114] このリード (read) 関数は、分散テーブルに対して、第 1 引数のKeyに対するリードアクセスを行い、その結果を引数としてMyCallbackのrunを呼び出すものである。ただし、このリード関数の記述は、リード関数が終了した時点では、分散テーブルに対するアクセスは終了しているとは限らない、という意味になる。

[0115] このように定義された意味の元、リード関数は次のように動作する。

リード関数は、引数に与えられたKeyとCallbackオブジェクトを、アクセス待ちリスト 104 に登録する。このときのアクセス待ちリスト 104 の例は、上記実施形態で示した図 2 のようになる。図 2 (a) の例のアクセス待ちリスト 104 は、与えられたKeyとCallbackを、Keyを担当する分散テーブルサーバ 10 ごとにまとめたものである。図 2 (b) の例のアクセス待ちリスト 104 は、与えられたKeyとCallbackを、Keyごとにまとめたものである。以下、特に、断らない限り、図 2 (a) のサーバ毎にまとめたアクセス待ちリスト 104 を用いるものとして説明する。

[0116] アクセス待ちリスト 104 に登録されたリードアクセス要求は、ある程度ためられ、決定部 210 によってその発行タイミングと発行先のサーバが決定される。決定部 210 の動作については、上記実施形態の決定部 108 と同様である。

[0117] 決定部 210 によって、ある分散テーブルサーバ 10 へのリードアクセス発行が決定されると、リードアクセス発行部 110 によって、アクセス待ちリスト 104 の当該サーバ欄にある少なくとも一つのキーに対するリードアクセスが、当該分散テーブルサーバ 10 に対して行われる。そして、リードアクセス発行部 110 は、それぞれのキーに対するバリューを受け取る。

[0118] そして、コールバック実行部 112 によって、キーと受け取ったバリューを用いて、アクセス待ちリスト 104 の当該サーバ欄に登録されていたコー

ルバックが実行される。上述したC++での例を用いると、コールバック実行部112は、Callbackオブジェクトのrunメンバ関数をKeyとValueを引数に実行する。

[0119] 以上により、クライアント装置200は、プログラム上離れた場所に存在する分散テーブルに対するリードアクセスにおいて、同一サーバに対する複数のアクセスをまとめることで、分散テーブルへのアクセスにおけるレイテンシの影響を削減することができる。

[0120] なお、コールバック内部ではさらにリード関数だけでなく、ライト関数またはアキュムレート関数の実行を行うこともでき、アクセス待ちリスト104に、同じキーに対するリード関数とともに、ライト関数またはアキュムレート関数をアクセス待ち処理として複数一緒に登録してもよい。また、同じキーに対するリード関数が複数回プログラムで実行された場合、アクセス待ちリスト104に複数登録されていてもよい。

[0121] 次に、分散テーブルへの書き込みアクセス（ライトアクセス）がある場合について説明する。

通常、共有されるデータに対し、複数のマシン（たとえば、クライアント装置200やクライアント装置200に接続される他のマシン）からアクセスされる場合には、ロック等の排他制御を行うが、本システムではそのような制御は対象としない。すなわち、本実施形態では、各マシンでの分散テーブルに対する読み込みおよび書き込みアクセスは任意の順番で実行できる場合を対象とする。

[0122] まず、クライアント装置200は、書き込みアクセスを、ライト（write）アクセスとアキュムレート（accumulate）アクセスに分類する。ライトアクセスは、キー（Key）に対応する値を上書きする通常の書き込みである。アキュムレートアクセスは、元々あった値と指定された値の間で演算を行い、その結果を格納するアクセスである。ここで、指定される演算は、結合法則が成り立つものであるとする。加算や乗算、最大値または最小値を求める演算がこの例となる。

[0123] たとえば、加算の場合、Key Xの値にYを加算するアクセスをaccumulate (add, X, Y) などとすると、Key Xの値が1の場合、accumulate (add, X, 1) , accumulate (add, X, 2) が発行されると、Key Xの値は4となる。この結果は、accumulate (add, X, 1) , accumulate (add, X, 2) が実行される順番に依存しない。

[0124] リードの場合と同様、クライアント装置200が受け付けるライトアクセス、およびアキュムレートアクセスはともに、個々のプログラムに記述されている、ライト関数、およびアキュムレート関数からそれぞれ呼び出されるものとする。リード関数の場合と同様、ライト関数、およびアキュムレート関数ともに、プログラム上において、その実行が終わった段階で、クライアント装置200を介した分散テーブルサーバ10へのライトアクセス、およびアキュムレートアクセスがそれぞれ終了しているとは限らない。

[0125] まず、ライトアクセスについて説明する。ライト (write) 関数をC++で記述した一例を下記に示す。

Write (Key, Value) ;

[0126] ライトアクセスは、その返り値を利用しないので、処理速度はレイテンシの影響を受けない。そのため、クライアント装置200は、ライトアクセスを基本的には無条件に発行してよい。ただし、プログラムによっては、同じキーに対するリード関数の実行がライト関数の実行の前に行われている場合、ライトがリードを追い越さないようにしたい場合も考えられる。これは同一マシン内においてであり、前述の通り、複数マシン間でのリードアクセスおよびライトアクセスの順序制御は考慮しない。そこで、本実施形態では、ライトアクセスがリードアクセスを追い越してよい場合と、追い越さない場合に分けて考える。

[0127] ライトアクセスがリードアクセスを追い越してよい場合、ライト関数実行部204により、ライト関数が実行されると、ライト関数実行部204がまずアクセス待ちリスト104を参照し、同一キーに対するリードがアクセス待ちリスト104に登録されているかどうかを調べる。もし登録されていれ

ば、ライト関数実行部204からの要求に従い、ライト／アキュムレートアクセス発行部208が、ライト要求をそのキーを担当するサーバに発行するとともに、そのキーとライト関数で書き込むバリューを引数にして、登録されていたコールバックを実行する。もし登録されていないならば、ライト関数実行部204からの要求に従い、ライト／アキュムレートアクセス発行部208が、単純にライト要求をそのキーを担当するサーバに発行する。

- [0128] つぎに、ライトアクセスがリードアクセスを追い越さないようにしたい場合を考える。本実施形態では、このような場合に対応するために、追い越されたくないリードにフラグ(true)を立てる。個々のプログラムでは、たとえば、先ほどのリード関数の引数を拡張し、以下のようにして、このキーに対するリードアクセスはライトに追い越されたくないことを示しておくことができる。これを以後追い越し不可のリードと呼ぶこととする。

```
read (Key, new MyCallback (...), true) ;
```

- [0129] このような指示がなされた場合、すなわち、追い越し不可フラグ(true)が指定された場合、登録部106(ライト関数実行部204)は、アクセス待ちリスト104に、その情報(追い越し不可フラグ(true))を保存する。すなわち、登録部106(ライト関数実行部204)は、図18に示すように、待機リードアクセスとしてキー(Key1)とコールバック(Callback1)と、さらに、追い越し不可フラグ(true)をアクセス待ちリスト104に登録する。

- [0130] そして、追い越し不可のリードアクセスと同じキー(Key1)に対するライト関数が実行された場合を考える。ライト関数実行部204によりライト関数が実行されると、ライト関数実行部204がまずアクセス待ちリスト104を参照する。同一キー(Key1)に対する追い越し不可のリードが存在した場合(フラグtrueが登録されている場合)、ライト関数実行部204(登録部106)が、当該ライトもアクセス待ちリスト104に登録する。この様子を図18に示す。

- [0131] 次に、ライト(write)がアクセス待ちリスト104に登録されている状態

(図 18 の状態)での各関数の動作を考える。

同一キーに対するリード関数が実行された場合、コールバック実行部 112 が、待ち合わせているライトのバリューを使ってコールバックを実行する。

同一キーに対するライト関数が実行された場合、ライト／アキュムレートアクセス発行部 208 が、待ち合わせているライトのバリューを新たに実行されたライト関数のバリューで上書きする。

同一キーに対するアキュムレート関数が実行された場合、ライト／アキュムレートアクセス発行部 208 が、ライト関数に指定されているバリューを元の値としてアキュムレート関数を実行した結果を計算し、ライト関数のバリューを変更する。たとえば、ライト関数に指定されているバリューが 1 で、アキュムレート関数が 2 を加算するものであった場合、ライト／アキュムレートアクセス発行部 208 が、ライト関数に指定するバリューを 3 に変更する。

[0132] そして、ライト (write) がアクセス待ちリスト 104 に登録されている状態 (図 18 の状態) で、リードアクセス発行部 110 により、アクセス待ちリスト 104 の当該リードアクセスが発行されれば、その完了を待ち、ライト／アキュムレートアクセス発行部 208 が、待ち合わせていたライトアクセスを発行する。

[0133] 次に、アキュムレートアクセスについて説明する。アキュムレート関数を C++ で記述した一例を下記に示す。

```
accumulate (TYPE, Key, Value) ;
```

ここで、第 1 引数の TYPE には実行する演算 (加算なら add, 乗算なら mul 等) が指定される。

[0134] まず、アキュムレート関数に指定されたキー (Key) について、追い越し不可のリードがアクセス待ちリスト 104 にない場合について説明する。この場合はアキュムレート関数実行部 206 からの要求に従い、ライト／アキュムレートアクセス発行部 208 が、アキュムレート関数を即時発行する。ラ

イト関数の場合と異なり、アクセス待ちリスト 104 に追い越し不可ではないリードアクセスがあるなしに関わらず、このとき、そのリードアクセスの Callback は実行しない。これは、現在のその Key に対応する Value がわからないと、アキュムレート関数の引数だけでは Value が決定しないためである。

[0135] 次に、追い越し不可のリードがアクセス待ちリスト 104 にある場合について説明する。この場合はライト関数の場合と同様、アキュムレート関数実行部 206 が当該アキュムレートアクセスをアクセス待ちリスト 104 に登録する。この様子を図 19 に示す。

[0136] アキュムレートがアクセス待ちリスト 104 に登録されている状態（図 19 の状態）での各関数の動作を考える。

同一キーに対するリード関数が実行された場合、リード関数実行部 202（登録部 106）が、（上述したライトアクセスの場合と異なり）さらにリードアクセスをアクセス待ちリスト 104 に登録する。この様子を図 20 に示す。この状態からさらに同一キーに対するリード、ライト、アキュムレートアクセスがあった場合は、これまで述べてきた場合と同様の動作を行う。

[0137] アキュムレートがアクセス待ちリスト 104 に登録されている状態で、同一キーに対するライト関数が実行された場合、ライト関数実行部 204（登録部 106）が、アクセス待ちリスト 104 のアキュムレートアクセスを当該ライトアクセスに置き換え、上書きする。

[0138] アキュムレートがアクセス待ちリスト 104 に登録されている状態で、同一キーに対するアキュムレート関数が実行された場合、アキュムレート関数実行部 206（登録部 106）が、同じ演算タイプ（たとえば、加算（add）同士など）なら、アキュムレートの Value を合算して更新する。たとえば、accumulate（add, Key1, 1）が登録されている状態で、accumulate（add, Key1, 2）が実行された場合、アキュムレート関数実行部 206（登録部 106）が、登録されているアキュムレートアクセスを accumulate（add, Key1, 3）に変更する。異なる演算タイプの場合は、アキュムレート関数実行部 206（登録部 106）が、アキュムレートアクセスをさらにその後に登録する（不図

示)。

[0139] アキュムレートがアクセス待ちリスト104に登録されている状態(図19の状態)で、リードアクセス発行部110により、アクセス待ちリスト104の当該リードアクセスが発行されれば、その完了を待ち、ライト／アキュムレートアクセス発行部208が、待ち合わせていたアキュムレートアクセスを発行する。ここで、待ち合わせていたアキュムレートアクセスに、さらにリードアクセスが待ち合わせていた場合(図20の状態)、アキュムレート後の値はリードアクセスによって計算できるため、コールバック実行部112が、この値を用いてさらに待ち合わせているリードアクセスのコールバックを実行する。

[0140] たとえば、図20の状態から、リードアクセス発行部110が、リードを発行することでKey1の値が1だと判明したとする。この値を用いて、コールバック実行部112がCallback1を実行する。登録されているアキュムレートのValue2が2だった場合、ライト／アキュムレートアクセス発行部208が、アキュムレートを発行するとともに、コールバック実行部112が、 $1 + 2 = 3$ を用いてCallback5を実行する。

[0141] また、決定部210による発行タイミングと発行先のサーバの決定処理手順については、上述したように第1～第7の方法が考えられる。ここでは、説明は省略する。

[0142] 次に、図21から図24のフローチャートを参照して本実施の形態の全体の動作について詳細に説明する。

図21は、本実施形態の分散システム1のクライアント装置200のリード関数実行部202の動作の一例を示すフローチャートである。

リード関数実行部202が、まずアクセス待ちリスト104を確認し、アクセス待ちリスト104において、同一キーに対して待つアクセスをアクセス待ちリスト104の後ろから調べる(ステップS201)。すなわち、アクセス待ちリスト104は図18から図20のように、ライトアクセスやアキュムレートアクセス、リードアクセスがその後に登録されている場合があ

るため、リード関数実行部202は、これを後から調べる。

- [0143] ライトアクセスが登録されている場合（ステップS203のYES：図18の状態）、コールバック実行部112が、登録されているバリューを用いて、リード関数の引数のコールバックを実行して（ステップS205）、本処理を終了する。このとき、アクセス待ちリスト104は図18の状態のままとなる。
- [0144] アキュムレートアクセスが登録されている場合（ステップS207のYES：図19の状態）、リード関数実行部202が、アキュムレートアクセスの後ろに引数に与えられたKeyとCallbackオブジェクトを登録する（ステップS209）。すなわち、図20のような状態になる。その後、リード関数実行部202は、決定部210に、リードアクセスを登録した旨通知する（ステップS213）。なお、決定部210がこの通知を必要としない場合は、このステップは省略できる。
- [0145] ライトアクセス、アキュムレートアクセスが登録されていない場合は（ステップS203のNO、かつ、ステップS207のNO）、リード関数実行部202は、引数に与えられたKeyとCallbackオブジェクトをアクセス待ちリスト104に登録する（ステップS211）。そして、リード関数実行部202は、決定部210に、リードアクセスを登録した旨通知する（ステップS213）。先ほどと同様、決定部210がこの通知を必要としない場合は、このステップは省略できる。また、リードアクセスを登録する際に、必要に応じて、決定部210は、その登録によって必要となるメモリ量をアクセス待ちリスト104の消費メモリ量に加算する。
- [0146] 前述の通り、リードアクセスのキーとコールバックが必要とするメモリ量を加算していくことでアクセス待ちリスト104の消費メモリ量を求めることができるし、登録毎に加算するメモリ量はプログラマが指定するようにしてもよい。決定部210は、後述のライト／アキュムレートアクセスを登録する際も同様に、メモリ量を加算していく。決定部210は、これらの場合は、ライト／アキュムレートアクセスのキーとバリューが必要とするメモリ

量（アキュムレートの場合はさらにTYPE情報に必要なメモリ量）を加算する。

[0147] 図22は、本実施形態の分散システム1のクライアント装置200のライト関数実行部204の動作一例を示すフローチャートである。

上記リード関数の場合と同様、ライト関数実行部204が、アクセス待ちリスト104において、同一キーに対して待つアクセスをアクセス待ちリスト104の後ろから調べる（ステップS221）。ライトアクセスが登録されている場合（ステップS223のYES：図18の状態）、ライト関数実行部204が、登録されているライトアクセスのバリューを引数に与えられたバリューで上書きする（ステップS225）。そして、ライト関数実行部204は、本処理を終了する。

[0148] アキュムレートアクセスが登録されている場合（ステップS227のYES：図19の状態）、ライト関数実行部204が、登録されているアキュムレートアクセスを当該ライトアクセスで上書きする（ステップS229）。すなわち、図20の状態になる。そして、ライト関数実行部204は、本処理を終了する。アキュムレートアクセスが登録されていない場合で（ステップS227のNO）、かつ、リードアクセスが登録されていなければ（ステップS231のNO）、ライト関数実行部204は、ライト／アキュムレートアクセス発行部208を用いライトアクセスを発行する（ステップS233）。そして、ライト関数実行部204は、本処理を終了する。

[0149] リードアクセスが登録されている場合（ステップS231のYES）、すなわち、図20の状態、あるいは図2の状態の場合、ライト関数実行部204は、登録されているリードアクセスが追い越し可能かどうかを調べる（ステップS235）。追い越し可能でなければ（ステップS235のNO）、すなわち、フラグtrueが登録されている場合、ライト関数実行部204は、当該ライトアクセスをアクセス待ちリスト104のこのリードアクセスの後ろに登録する（ステップS237）。追い越し可能であった場合（ステップS235のYES）、ライト関数実行部204は、ライト関数の引数のバリ

ユーを用いて当該リードアクセスのコールバックをコールバック実行部 112 によって実行させ、当該リードアクセスをアクセス待ちリスト 104 から削除する（ステップ S 239）。さらに、アクセス待ちリスト 104 の同一キーに対して待つアクセスの 1 つ前の要素に注目して、ステップ S 221 に戻り、ライト関数実行部 204 は、上記と同様な処理を繰り返す（ステップ S 241）。アクセス待ちリスト 104 の同一キーに対して待つアクセスの全ての要素について処理が終了したら、ライト関数実行部 204 は、最後に空の要素に対して処理を繰り返した後、本処理を終了する。

[0150] たとえば、図 20 の状態から Key1 を引数に持つライト関数を実行した場合（ステップ S 231 の YES）、アクセス待ちリスト 104 の末尾にあるリードアクセスは追い越し可能であるため（ステップ S 235 の YES）、ライト関数実行部 204 は、コールバック実行部 112 により当該リードアクセスの Callback5 を実行させ、当該リードアクセスをアクセス待ちリスト 104 から取り除く（ステップ S 239）。すると、アクセス待ちリスト 104 は、図 19 の状態になる。そして、ステップ S 221 に戻り、アクセス待ちリスト 104 の末尾のアクムレートアクセスに注目して、ライト関数実行部 204 は、処理を繰り返す（ステップ S 241）。

[0151] また、図 2 の状態から Key1 を引数に持つライト関数を実行した場合（ステップ S 231 の YES）、リードアクセスは追い越し可能であるため（ステップ S 235 の YES）、ライト関数実行部 204 は、コールバック実行部 112 により当該リードアクセスのコールバックを実行させ、当該アクセス待ちリスト 104 から削除する（ステップ S 239）。次に注目する要素がアクセス待ちリスト 104 にないので、ライト関数実行部 204 は、空の要素に対して次の繰り返し処理を行う。ここでは、ライトアクセス、アクムレートアクセス、リードアクセスのいずれも登録されていないため、ライト関数実行部 204 は、最終的にライト／アクムレートアクセス発行部 208 を用いライトアクセスを発行して処理を終了する（ステップ S 237）。

[0152] 図 23 は、本実施形態の分散システム 1 のクライアント装置 200 のアキ

キュムレート関数実行部206の動作一例を示すフローチャートである。

ライト関数の場合と同様、アキュムレート関数実行部206が、アクセス待ちリスト104において、同一キーに対して待つアクセスをアクセス待ちリスト104の後ろから調べる（ステップS251）。ライトアクセスが登録されている場合（ステップS253のYES：図18の状態）、アキュムレート関数実行部206は、登録されているライトアクセスのバリューを元にアキュムレート演算を実行し、その結果で登録されているライトアクセスのバリューを上書きする（ステップS255）。たとえば、ライト関数に指定されているバリューが1で、アキュムレート関数が2を加算するものであった場合、アキュムレート関数実行部206は、ライト関数に指定するバリューを3に変更する。そして、アキュムレート関数実行部206は、本処理を終了する。

[0153] アキュムレートアクセスが登録されている場合（ステップS257のYES：図19の状態）、アキュムレート関数実行部206が、まず演算タイプが登録されているものと同じかどうかを調べる（ステップS259）。演算タイプが同じでなければ（ステップS259のNO）、アキュムレート関数実行部206は、アキュムレートアクセスをアクセス待ちリスト104の後ろに登録する（ステップS261）。そして、アキュムレート関数実行部206は、本処理を終了する。演算タイプが同じであれば（ステップS259のYES）、アキュムレート関数実行部206は、登録されているアキュムレートアクセスのバリューと、関数の引数のバリューを元に、アキュムレート演算を行い、その結果で登録されているアキュムレートアクセスのバリューを上書きする（ステップS263）。そして、アキュムレート関数実行部206は、本処理を終了する。

[0154] たとえば、accumulate (add, Key1, 1) が登録されている状態でaccumulate (add, Key1, 2) が実行された場合（ステップS257のYES、かつステップS259のYES）、アキュムレート関数実行部206は、登録されているアキュムレートアクセスをaccumulate (add, Key1, 3) に変更する（ステッ

プS 2 6 3)。

[0155] また、リードアクセスが登録されていなければ(ステップS 2 6 5のNO)、アキュムレート関数実行部2 0 6は、ライト／アキュムレートアクセス発行部2 0 8を用いアキュムレートアクセスを発行する(ステップS 2 7 3)。そして、アキュムレート関数実行部2 0 6は、本処理を終了する。リードアクセスが登録されている場合(ステップS 2 6 5のYES)、すなわち、図2 0の状態、あるいは図2の状態の場合、アキュムレート関数実行部2 0 6が、このリードアクセスが追い越し可能かどうかを調べる(ステップS 2 6 7)。

[0156] 追い越し可能でなければ(ステップS 2 6 7のNO)、アキュムレート関数実行部2 0 6は、当該アキュムレートアクセスをこのリードアクセスの後ろに登録する(ステップS 2 6 1)。そして、アキュムレート関数実行部2 0 6は、本処理を終了する。追い越し可能であれば(ステップS 2 6 7のYES)、アキュムレート関数実行部2 0 6は、アクセス待ちリスト1 0 4の同一キーに対して待つアクセスの1つ前の要素についてステップS 2 5 1に戻り、上記と同様な処理を繰り返す(ステップS 2 7 1)。アクセス待ちリスト1 0 4の同一キーに対して待つアクセスの全ての要素について処理が終了したら、アキュムレート関数実行部2 0 6は、最後に空の要素に対して処理を繰り返した後、本処理を終了する。

[0157] たとえば、図2 0の状態からKey1を引数に持つアキュムレート関数を実行した場合(ステップS 2 6 5のYES)、アクセス待ちリスト1 0 4の末尾にあるリードアクセスは追い越し可能であるため(ステップS 2 6 7のYES)、アキュムレート関数実行部2 0 6は、アクセス待ちリスト1 0 4の末尾1つ前のアキュムレートアクセスに注目して処理を繰り返す(ステップS 2 7 1)。アクセス待ちリスト1 0 4の同一キーに対して待つアクセスの全ての要素について処理が終了したら、アキュムレート関数実行部2 0 6は、最後に空の要素に対して処理を繰り返した後、本処理を終了する。

[0158] 図2の状態からアキュムレート関数を実行した場合(ステップS 2 6 5の

YES、かつ、ステップS 2 6 7のNO)、次に注目する要素がアクセス待ちリスト1 0 4にないので、アキュムレート関数実行部2 0 6は、空の要素に対して次の繰り返し処理を行う。ここでは、ライトアクセス、アキュムレートアクセス、リードアクセスのいずれも登録されていないため(ステップS 2 5 3のNO、かつ、ステップS 2 5 7のNO、かつ、ステップS 2 6 5のNO)、アキュムレート関数実行部2 0 6は、最終的にライト／アキュムレートアクセス発行部2 0 8を用いてアキュムレートアクセスを発行して処理を終了する(ステップS 2 7 3)。

[0159] 図2 4は、本実施形態の分散システム1のクライアント装置2 0 0のリードアクセス発行部1 1 0の動作の一例を示すフローチャートである。ライト／アキュムレートアクセス発行部2 0 8は、単純にライトアクセス、アキュムレートアクセスを担当するサーバに発行するだけであるが、リードアクセス発行部1 1 0は、対象とするサーバを指定して呼び出され、図2 4のような動作を行う。

[0160] まず、リードアクセス発行部1 1 0は、指定されたサーバに対し、アクセス待ちリスト1 0 4の先頭から参照し、先頭の待機リードアクセスから順に以下の処理を繰り返し実行する(ステップS 2 8 1)。リードアクセス発行部1 1 0は、複数のリードアクセスについて、それぞれリードアクセスを順に発行、各キーに対応するバリューを受け取る(ステップS 2 8 3)。決定部2 1 0で必要とされる場合は、リードアクセス発行部1 1 0は、この発行時刻をアクセス待ちリスト1 0 4に記録しておく。

そして、リードアクセス発行部1 1 0は、得られたキーとバリューを用い、登録されていたコールバックを実行し、アクセス待ちリスト1 0 4から削除する(ステップS 2 8 5)。アクセス待ちリスト1 0 4の後ろに、ステップS 2 8 3で発行したキーのリードアクセスの発行を待っているライトアクセスが存在した場合(ステップS 2 8 7のYES)、リードアクセス発行部1 1 0は、ライト／アキュムレートアクセス発行部2 0 8を用いてライトアクセスを発行する(ステップS 2 8 9)。

[0161] 同様にアキュムレートアクセスが存在した場合（ステップS 2 9 1のYES）、リードアクセス発行部1 1 0は、ライト／アキュムレートアクセス発行部2 0 8を用いてアキュムレートアクセスを発行する（ステップS 2 9 3）。この場合、リードアクセス発行部1 1 0は、リードアクセスの発行によって得られたキーとバリューを用い、アキュムレートアクセス後のキーに対応するバリューを計算する（ステップS 2 9 5）。なお、ライトアクセスもアキュムレートアクセスも存在しない場合（ステップS 2 8 7のNO、かつ、ステップS 2 9 1のNO）、リードアクセス発行部1 1 0は、ステップS 2 8 3で受け取ったキーとバリューを用いて計算する。

そして、リードアクセス発行部1 1 0は、このキーとバリューの値を用い、後にある当該キーのリード／ライト／アキュムレートアクセスについて（ステップS 2 9 7のYES）、処理を繰り返し実行する（ステップS 2 9 8）。存在しない場合（ステップS 2 9 7のNO）、リードアクセス発行部1 1 0は、本処理を終了する。

なお、繰り返し処理において、リードアクセスが後に存在する場合（ステップS 2 9 9のYES）、ステップS 2 8 5に戻り、リードアクセスが存在しない場合（ステップS 2 9 9のNO）、ステップS 2 8 7に戻ることとなる。

[0162] たとえば、図20の状態から分散テーブルサーバS 1に対してリードアクセスを発行したとすると、リードアクセス発行部1 1 0は、得られたKey1に対するバリューを用いてアキュムレート処理を実行し、その演算後の値がKey1に対するバリューとして得られたとして、後続のリードアクセスについて繰り返し処理を行う。

[0163] 次に、本実施形態のクライアント装置2 0 0の決定部2 1 0によって決定されるアクセス発行処理の動作について、以下説明する。上記実施形態の決定部1 0 8の処理手順を示した図7～図16を用いて、第1～第7の方法についてそれぞれ説明する。決定部2 1 0は、基本的に上記実施形態の決定部1 0 8と同様に動作する。

[0164] 図7に示す第1の方法の決定処理では、まず、決定部210が、リード関数からアクセス待ちリスト104に登録した通知を受け取る（ステップS121）。この通知を受け取ると（ステップS121のYES）、本処理が開始し、決定部210が、アクセス待ちリスト104を参照し、担当するリードアクセスの数が一定数以上のサーバがあるかを調べる（ステップS123）。もしあれば、リードアクセス発行部110が、当該サーバに対するリードアクセスを発行する（ステップS125）。なお、このリードアクセス発行部110の動作は、実際には、図24を用いて説明した上記動作となるが、以下詳細な説明は省略し、単にリードアクセスを発行するものとして説明する。

[0165] 図8に示す第2の方法の決定処理では、決定部210が、まず、リード関数からアクセス待ちリスト104に登録した通知を受け取る（ステップS121）。この通知を受け取ると（ステップS121のYES）、本処理が開始し、決定部210が、アクセス待ちリスト104で消費されているメモリ量が一定量以上かを調べる（ステップS131）。もし一定量以上であれば（ステップS131のYES）、リードアクセス発行部110が、最もアクセス待ち数が多いサーバに対してリードアクセスを発行する（ステップS133）。

[0166] 図9に示す第3の方法の決定処理では、決定部210が、まず、リード関数からアクセス待ちリスト104に登録した通知を受け取る（ステップS121）。この通知を受け取ると（ステップS121のYES）、本処理が開始し、決定部210が、アクセス待ちリスト104で消費されているメモリ量が一定量以上かを調べる（ステップS131）。もし一定量以上であれば（ステップS131のYES）、リードアクセス発行部110が、1つのキーあたりにかかるバリュウ取得時間が最も少ないサーバに対してリードアクセスを発行する（ステップS141）。1つのキーあたりにかかるバリュウ取得時間は、前述の式（1）で求めることができる。

[0167] 図10～図12に示す第4の方法の決定処理では、決定部210が、リー

ド関数からアクセス待ちリスト104に登録した通知を受け取る（図10のステップS121のYES）、あるいはリード関数からアクセス待ちリスト104に登録した通知が所定回数以上来る（図11のステップS121のYES、かつ、ステップS161のYES）、あるいはタイマから一定時間ごとに通知をうけとる（図12のステップS171のYES）、のいずれかを契機として、処理を開始する。そして、決定部210が、アクセス待ちリスト104を参照し、1つのキーあたりにかかるバリュウ取得時間の最小値が閾値以下のサーバがあるかを調べる（ステップS151）。もしあれば（ステップS151のYES）、リードアクセス発行部110が、当該サーバに対するリードアクセスを発行する（ステップS125）。

[0168] 図13に示す第5の方法の決定処理では、決定部210が、タイマから一定時間ごとに通知を受け取り（ステップS171のYES）、一定時間以上リードアクセスが発行されていないサーバがあるか、アクセス待ちリスト104に記録した前回のリード発行時刻を元に調べる（ステップS181）。もしそのようなサーバがあれば（ステップS181のYES）、リードアクセス発行部110が、当該サーバに対するリードアクセスを発行する（ステップS125）。

[0169] 図14に示す第6の方法の決定処理では、決定部210が、プログラムから明示的にリードアクセスを発行するサーバを指定した関数を呼び出す（ステップS191）。そして、リードアクセス発行部110が、指定されたタイミングで、指定されたサーバに対して、リードアクセスを発行する（ステップS195）。

[0170] 図15および図16に示す第7の方法の決定処理では、決定部210が、リード関数からアクセス待ちリスト104に登録した通知を受け取る（図15のステップS121）。この通知を受け取ると（図15のステップS121のYES）、決定部210が、アクセス待ちリスト104で消費されているメモリ量が一定量以上かを調べる（図15のステップS131）。もし一定量以上であれば（図15のステップS131のYES）、リードアクセス

発行部 110 が、1つのキーあたりにかかるバリュ取得時間が最も少ないサーバに対してリードアクセスを発行する（図 15 のステップ S 141）。そうでなければ（図 15 のステップ S 131 の NO）、決定部 210 が、リード関数からアクセス待ちリストに登録した通知が所定回数以上来たかどうかを調べる（図 15 のステップ S 161）。なお、本ステップは省略して、成立したものとして次に進んでもよい。

[0171] もし所定回数以上来ていれば（図 15 のステップ S 161 の YES）、決定部 210 が、1つのキーあたりにかかるバリュ取得時間の最小値が閾値以下のサーバがあるかを調べる（図 15 のステップ S 151）。もしあれば（図 15 のステップ S 151 の YES）、リードアクセス発行部 110 が、当該サーバに対するリードアクセスを発行する（図 15 のステップ S 125）。

[0172] また、本方法では、決定部 210 が、タイマから一定時間ごとに通知を受け取り（図 16 のステップ S 171 の YES）、一定時間以上リードアクセスが発行されていないサーバがあるか、アクセス待ちリスト 104 に記録した前回のリード発行時刻を元に調べる（図 16 のステップ S 181）。もしそのようなサーバがあれば（図 16 のステップ S 181 の YES）、リードアクセス発行部 110 が、当該サーバに対するリードアクセスを発行する（図 16 のステップ S 125）。もしなければ（図 16 のステップ S 181 の NO）、決定部 210 が、1つのキーあたりにかかるバリュ取得時間の最小値が閾値以下のサーバがあるかを調べる（図 16 のステップ S 151）。もしあれば（図 16 のステップ S 151 の YES）、リードアクセス発行部 110 が、当該サーバに対するリードアクセスを発行する（図 16 のステップ S 125）。

[0173] また、本方法では、図 14 に示すように、決定部 210 が、プログラムから明示的にリードアクセスを発行するサーバを指定した関数を呼び出してもよい（ステップ S 191）。そして、リードアクセス発行部 110 が、指定されたタイミングで指定されたサーバに対して、リードアクセスを発行する

(ステップS 1 9 5)。

[0174] 以上説明したように、本発明の実施の形態の分散システム 1 によれば、上記実施形態と同様な効果を奏するとともに、ライトアクセスやアキュムレートアクセスとともに、効率のよい処理を行うことができる。

[0175] 以上、図面を参照して本発明の実施形態について述べたが、これらは本発明の例示であり、上記以外の様々な構成を採用することもできる。

[0176] たとえば、本発明の実施形態における最小構成は、図 2 7 に示すような情報処理装置 1 0 0 0 とすることができる。情報処理装置 1 0 0 0 は、複数のサーバ（分散テーブルサーバ 1 0）がそれぞれ有する複数の分散テーブル（不図示）からデータを読み出すリード（read）アクセス処理のキー（Key）、および、キーに基づき読み出した結果を用いて実行する処理を定めるコールバック（Callback）、を受け付ける受付部 1 0 2 と、受け付けたリードアクセス処理を待機させ、該当する分散テーブルサーバ 1 0 に後で発行する待機リードアクセスとしてキーとコールバックをアクセス待ちリスト 1 0 4 に登録する登録部 1 0 6 と、アクセス待ちリスト 1 0 4 に登録された待機リードアクセスについて、所定の条件に従い、待機リードアクセスを発行する発行先の分散テーブルサーバ 1 0 および発行タイミングを決定する決定部 1 0 8 と、決定部 1 0 8 が決定した発行先の分散テーブルサーバ 1 0 に、決定した発行タイミングで、待機リードアクセスを発行し、分散テーブルサーバ 1 0 から当該アクセス結果を受け取るリードアクセス発行部 1 1 0 と、リードアクセス発行部 1 1 0 が受け取ったアクセス結果（バリュー）を用いて、アクセス待ちリスト 1 0 4 に登録された該当する待機リードアクセスのコールバックを実行し、実行したコールバックの待機リードアクセスをアクセス待ちリスト 1 0 4 から削除するコールバック実行部 1 1 2 と、を備える。

[0177] 図 2 7 の情報処理装置 1 0 0 0 は、上記実施形態のクライアント装置を実現する図 3 のコンピュータ 6 0 と同様なコンピュータのハードウェアとソフトウェアの任意の組み合わせによって実現される。図 2 7 は、ハードウェア単位の構成ではなく、論理的な機能単位のブロックを示している。

本実施の形態の情報処理装置１０００では、コンピュータプログラムに対応する各種の処理動作をコンピュータのＣＰＵが実行することにより、図２７の各種ユニットが各種機能として実現される。

[0178] この構成によれば、情報処理装置１０００が、任意のプログラムを対象として、プログラム上の任意の場所に現れる分散テーブルサーバ１０へのリード関数による複数のリードアクセス処理を、リードアクセス毎に行うのではなく、アクセス待ちリスト１０４に登録して一定数以上待機させることで、サーバ毎にまとめて実行することができる。そして、この構成により、プログラム上に複数現れるリードアクセス処理を、その都度処理する場合に比較して、リードアクセス時にかかるレイテンシによるアクセス処理速度への影響を削減することができ、高速化を図ることができる。

[0179] 以上、実施形態および実施例を参照して本願発明を説明したが、本願発明は上記実施形態および実施例に限定されるものではない。本願発明の構成や詳細には、本願発明の範囲内で当業者が理解し得る様々な変更をすることができる。

[0180] 以下、参考形態の例を付記する。

１． 複数のサーバがそれぞれ有する複数の分散テーブルからデータを読み出すリードアクセス処理のキー、および、前記キーに基づき読み出した結果を用いて実行する処理を定めるコールバック、を受け付ける受付手段と、

受け付けた前記リードアクセス処理を待機させ、該当するサーバに後で発行する待機リードアクセスとして前記キーと前記コールバックをアクセス待ちリストに登録する登録手段と、

前記アクセス待ちリストに登録された前記待機リードアクセスについて、所定の条件に従い、前記待機リードアクセスを発行する発行先のサーバおよび発行タイミングを決定する決定手段と、

前記決定手段が決定した前記発行先の前記サーバに、決定した前記発行タイミングで、前記待機リードアクセスを発行し、前記サーバから当該アクセス結果を受け取るリードアクセス発行手段と、

前記リードアクセス発行手段が受け取った前記アクセス結果を用いて、前記アクセス待ちリストに登録された該当する待機リードアクセスの前記コールバックを実行し、実行した前記コールバックの前記待機リードアクセスを前記アクセス待ちリストから削除するコールバック実行手段と、を備える分散システム。

2. 1. に記載の分散システムにおいて、

前記登録手段は、前記サーバ毎、または前記キー毎に、前記待機リードアクセスを分類して前記アクセス待ちリストに登録し、

前記決定手段は、前記サーバ毎または前記キー毎に、前記待機リードアクセスの前記発行タイミングを決定する分散システム。

3. 1. または2に記載の分散システムにおいて、

前記決定手段は、各前記サーバの性能情報に基づいて求まる、前記キー1つあたりにかかる前記アクセス結果の取得時間に基づいて、前記リードアクセスの前記発行先となるサーバを選択する分散システム。

4. 1. 乃至3. いずれかに記載の分散システムにおいて、

前記決定手段は、前記アクセス待ちリストに登録された前記待機リードアクセスの件数が一定数以上になったときを、前記発行タイミングであると決定する分散システム。

5. 1. 乃至4. いずれかに記載の分散システムにおいて、

前記決定手段は、

前記アクセス待ちリストのメモリ量が一定量以上になった場合、前記発行タイミングになったと判断し、

前記メモリ量が一定量以上になった時点で、前記アクセス待ちリストに登録された前記待機リードアクセスの件数が多いサーバを前記リードアクセスの前記発行先として選択する分散システム。

6. 1. 乃至5. いずれかに記載の分散システムにおいて、

前記決定手段は、一定時間以上、前記リードアクセス発行手段により前記リードアクセスが発行されていないサーバを前記リードアクセスの前記発行

先として選択する分散システム。

7. 1. 乃至 6. いずれかに記載の分散システムにおいて、

前記受付手段は、前記分散テーブルのキーに対して値を書き込むライトアクセス処理をさらに受け付け、

前記アクセス待ちリストを参照し、前記ライトアクセス処理と同じキーに対する前記待機リードアクセスが存在する場合、前記コールバック実行手段は、前記ライトアクセス処理の前記値で前記待機リードアクセスの前記コールバックを実行し、前記アクセス待ちリストから当該待機リードアクセスを削除する分散システム。

8. 7. に記載の分散システムにおいて、

前記受付手段は、前記リードアクセス処理より後に受け付ける前記ライトアクセス処理を、先に処理することを禁止する禁止フラグを、前記リードアクセス処理の前記キーと前記コールバックとともに受け付け、

前記登録手段は、受け付けた前記禁止フラグを前記待機リードアクセスの前記キーおよび前記コールバックとともに前記アクセス待ちリストに登録し、

さらに、前記登録手段は、前記アクセス待ちリストを参照し、受け付けた前記ライトアクセス処理と同じキーに対する前記待機リードアクセスの前記禁止フラグが登録されている場合、受け付けた前記ライトアクセス処理を待機させ、該当するサーバに後で発行する待機ライトアクセスとして前記キーと前記値を前記アクセス待ちリストに登録する分散システム。

9. 3. に記載の分散システムにおいて、

前記決定手段は、前記キー1つあたりにかかる前記アクセス結果の取得時間を前記サーバ毎に算出し、最小値が閾値以下のサーバを前記リードアクセスの前記発行先として選択する分散システム。

10. 9. に記載の分散システムにおいて、

前記決定手段は、前記登録手段が、前記アクセス待ちリストに前記待機リードアクセスを登録する度に、前記キー1つあたりにかかる前記アクセス結

果の取得時間を算出し、算出される度に、前記発行先の選択を行う分散システム。

11. 9. に記載の分散システムにおいて、

前記決定手段は、前記登録手段による前記アクセス待ちリストへの前記待機リードアクセスの登録が所定回数毎に、前記キー1つあたりにかかる前記アクセス結果の取得時間を算出し、算出される度に、前記発行先の選択を行う分散システム。

12. 9. に記載の分散システムにおいて、

前記決定手段は、一定時間毎に、前記キー1つあたりにかかる前記アクセス結果の取得時間を算出し、算出される度に、前記発行先の選択を行う分散システム。

13. 1. 乃至12. いずれかに記載の分散システムにおいて、

前記決定手段は、前記リードアクセスの前記発行先と、前記発行タイミングの指定を受け付け、前記指定に従い、前記リードアクセスの前記発行先と、前記発行タイミングを決定する分散システム。

[0181] 14. 情報処理装置が、

複数のサーバがそれぞれ有する複数の分散テーブルからデータを読み出すリードアクセス処理のキー、および、前記キーに基づき読み出した結果を用いて実行する処理を定めるコールバック、を受け付け、

受け付けた前記リードアクセス処理を待機させ、該当するサーバに後で発行する待機リードアクセスとして前記キーと前記コールバックをアクセス待ちリストに登録し、

前記アクセス待ちリストに登録された前記リードアクセスについて、所定の条件に従い、前記待機リードアクセスを発行するサーバおよび発行タイミングを決定し、

決定した前記サーバに、決定した前記発行タイミングで、前記待機リードアクセスを発行し、前記サーバから当該アクセス結果を受け取り、

受け取った前記アクセス結果を用いて、前記アクセス待ちリストに登録さ

れた該当する待機リードアクセスの前記コールバックを実行し、実行した前記コールバックの前記待機リードアクセスを前記アクセス待ちリストから削除する情報処理装置のデータ処理方法。

15. 14. に記載の情報処理装置のデータ処理方法において、
前記情報処理装置が、

前記リードアクセス処理を受け付けたとき、前記サーバ毎、または前記キー毎に、前記待機リードアクセスを分類して前記アクセス待ちリストに登録し、

前記アクセス待ちリストに登録された前記リードアクセスについて、前記サーバ毎または前記キー毎に、前記待機リードアクセスの前記発行タイミングを決定する情報処理装置のデータ処理方法。

16. 14. または 15. に記載の情報処理装置のデータ処理方法において、

前記情報処理装置が、

各前記サーバの性能情報に基づいて求まる、前記キー 1 つあたりにかかる前記アクセス結果の取得時間に基づいて、前記リードアクセスの前記発行先となるサーバを選択する情報処理装置のデータ処理方法。

17. 14. 乃至 16. いずれかに記載の情報処理装置のデータ処理方法において、

前記情報処理装置が、

前記アクセス待ちリストに登録された前記待機リードアクセスの件数が一定数以上になったときを、前記発行タイミングであると決定する情報処理装置のデータ処理方法。

18. 14. 乃至 17. いずれかに記載の情報処理装置のデータ処理方法において、

前記情報処理装置が、

前記アクセス待ちリストのメモリ量が一定量以上になった場合、前記発行タイミングになったと判断し、

前記メモリ量が一定量以上になった時点で、前記アクセス待ちリストに登録された前記待機リードアクセスの件数が多いサーバを前記リードアクセスの前記発行先として選択する情報処理装置のデータ処理方法。

19. 14. 乃至 18. いずれかに記載の情報処理装置のデータ処理方法において、

前記情報処理装置が、

一定時間以上、前記リードアクセスが発行されていないサーバを前記リードアクセスの前記発行先として選択する情報処理装置のデータ処理方法。

20. 14. 乃至 19. いずれかに記載の情報処理装置のデータ処理方法において、

前記情報処理装置が、

前記分散テーブルのキーに対して値を書き込むライトアクセス処理をさらに受け付け、

前記アクセス待ちリストを参照し、前記ライトアクセス処理と同じキーに対する前記待機リードアクセスが存在する場合、前記ライトアクセス処理の前記値で前記待機リードアクセスの前記コールバックを実行し、前記アクセス待ちリストから当該待機リードアクセスを削除する情報処理装置のデータ処理方法。

21. 20. に記載の情報処理装置のデータ処理方法において、

前記情報処理装置が、

前記リードアクセス処理より後に受け付ける前記ライトアクセス処理を、先に処理することを禁止する禁止フラグを、前記リードアクセス処理の前記キーと前記コールバックとともに受け付け、

受け付けた前記禁止フラグを前記待機リードアクセスの前記キーおよび前記コールバックとともに前記アクセス待ちリストに登録し、

さらに、前記アクセス待ちリストを参照し、受け付けた前記ライトアクセス処理と同じキーに対する前記待機リードアクセスの前記禁止フラグが登録されている場合、受け付けた前記ライトアクセス処理を待機させ、該当する

サーバに後で発行する待機ライトアクセスとして前記キーと前記値を前記アクセス待ちリストに登録する情報処理装置のデータ処理方法。

22. 16. に記載の情報処理装置のデータ処理方法において、

前記情報処理装置が、

前記キー1つあたりにかかる前記アクセス結果の取得時間を前記サーバ毎に算出し、最小値が閾値以下のサーバを前記リードアクセスの前記発行先として選択する情報処理装置のデータ処理方法。

23. 22. に記載の情報処理装置のデータ処理方法において、

前記情報処理装置が、

前記アクセス待ちリストに前記待機リードアクセスに登録する度に、前記キー1つあたりにかかる前記アクセス結果の取得時間を算出し、算出される度に、前記発行先の選択を行う情報処理装置のデータ処理方法。

24. 22. に記載の情報処理装置のデータ処理方法において、

前記情報処理装置が、

前記アクセス待ちリストへの前記待機リードアクセスの登録が所定回数毎に、前記キー1つあたりにかかる前記アクセス結果の取得時間を算出し、算出される度に、前記発行先の選択を行う情報処理装置のデータ処理方法。

25. 22. に記載の情報処理装置のデータ処理方法において、

前記情報処理装置が、

一定時間毎に、前記キー1つあたりにかかる前記アクセス結果の取得時間を算出し、算出される度に、前記発行先の選択を行う情報処理装置のデータ処理方法。

26. 14. 乃至25. いずれかに記載の情報処理装置のデータ処理方法において、

前記情報処理装置が、

前記リードアクセスの前記発行先と、前記発行タイミングの指定を受け付け、前記指定に従い、前記リードアクセスの前記発行先と、前記発行タイミングを決定する情報処理装置のデータ処理方法。

[0182] 27. 情報処理装置を実現するコンピュータに、

複数のサーバがそれぞれ有する複数の分散テーブルからデータを読み出す
リードアクセス処理のキー、および、前記キーに基づき読み出した結果を用
いて実行する処理を定めるコールバック、を受け付ける手順、

受け付けた前記リードアクセス処理を待機させ、該当するサーバに後で発
行する待機リードアクセスとして前記キーと前記コールバックをアクセス待
ちリストに登録する手順、

前記アクセス待ちリストに登録された前記待機リードアクセスについて、
所定の条件に従い、前記待機リードアクセスを発行するサーバおよび発行タ
イミングを決定する手順、

決定した前記サーバに、決定した前記発行タイミングで、前記待機リード
アクセスを発行し、前記サーバから当該アクセス結果を受け取る手順、

受け取った前記アクセス結果を用いて、前記アクセス待ちリストに登録さ
れた該当する待機リードアクセスの前記コールバックを実行し、実行した前
記コールバックの前記待機リードアクセスを前記アクセス待ちリストから削
除する手順、を実行させるためのプログラム。

28. 27. に記載のプログラムにおいて、

前記リードアクセス処理を受け付けたとき、前記サーバ毎、または前記キ
ー毎に、前記待機リードアクセスを分類して前記アクセス待ちリストに登録
する手順、

前記アクセス待ちリストに登録された前記リードアクセスについて、前記
サーバ毎または前記キー毎に、前記待機リードアクセスの前記発行タイミン
グを決定する手順をコンピュータに実行させるためのプログラム。

29. 27. または28. に記載のプログラムにおいて、

各前記サーバの性能情報に基づいて求まる、前記キー1つあたりにかかる
前記アクセス結果の取得時間に基づいて、前記リードアクセスの前記発行先
となるサーバを選択する手順をコンピュータに実行させるためのプログラム
。

30. 27. 乃至 29. いずれかに記載のプログラムにおいて、

前記アクセス待ちリストに登録された前記待機リードアクセスの件数が一定数以上になったときを、前記発行タイミングであると決定する手順をコンピュータに実行させるためのプログラム。

31. 27. 乃至 30. いずれかに記載のプログラムにおいて、

前記アクセス待ちリストのメモリ量が一定量以上になった場合、前記発行タイミングになったと判断する手順、

前記メモリ量が一定量以上になった時点で、前記アクセス待ちリストに登録された前記待機リードアクセスの件数が多いサーバを前記リードアクセスの前記発行先として選択する手順をコンピュータに実行させるためのプログラム。

32. 27. 乃至 31. いずれかに記載のプログラムにおいて、

一定時間以上、前記リードアクセスが発行されていないサーバを前記リードアクセスの前記発行先として選択する手順をコンピュータに実行させるためのプログラム。

33. 27. 乃至 32. いずれかに記載のプログラムにおいて、

前記分散テーブルのキーに対して値を書き込むライトアクセス処理をさらに受け付ける手順、

前記アクセス待ちリストを参照し、前記ライトアクセス処理と同じキーに対する前記待機リードアクセスが存在する場合、前記ライトアクセス処理の前記値で前記待機リードアクセスの前記コールバックを実行する手順、

前記アクセス待ちリストから当該待機リードアクセスを削除する手順をコンピュータに実行させるためのプログラム。

34. 33. に記載のプログラムにおいて、

前記リードアクセス処理より後に受け付ける前記ライトアクセス処理を、先に処理することを禁止する禁止フラグを、前記リードアクセス処理の前記キーと前記コールバックとともに受け付ける手順、

受け付けた前記禁止フラグを前記待機リードアクセスの前記キーおよび前

記コールバックとともに前記アクセス待ちリストに登録する手順、

前記アクセス待ちリストを参照し、受け付けた前記ライトアクセス処理と同じキーに対する前記待機リードアクセスの前記禁止フラグが登録されている場合、受け付けた前記ライトアクセス処理を待機させ、該当するサーバに後で発行する待機ライトアクセスとして前記キーと前記値を前記アクセス待ちリストに登録する手順をコンピュータに実行させるためのプログラム。

35. 29. に記載のプログラムにおいて、

前記キー1つあたりにかかる前記アクセス結果の取得時間を前記サーバ毎に算出し、最小値が閾値以下のサーバを前記リードアクセスの前記発行先として選択する手順をコンピュータに実行させるためのプログラム。

36. 35. に記載のプログラムにおいて、

前記アクセス待ちリストに前記待機リードアクセスを登録する度に、前記キー1つあたりにかかる前記アクセス結果の取得時間を算出し、算出される度に、前記発行先の選択を行う手順をコンピュータに実行させるためのプログラム。

37. 35. に記載のプログラムにおいて、

前記アクセス待ちリストへの前記待機リードアクセスの登録が所定回数毎に、前記キー1つあたりにかかる前記アクセス結果の取得時間を算出し、算出される度に、前記発行先の選択を行う手順をコンピュータに実行させるためのプログラム。

38. 35. に記載のプログラムにおいて、

一定時間毎に、前記キー1つあたりにかかる前記アクセス結果の取得時間を算出し、算出される度に、前記発行先の選択を行う手順をコンピュータに実行させるためのプログラム。

39. 27. 乃至38. いずれかに記載のプログラムにおいて、

前記リードアクセスの前記発行先と、前記発行タイミングの指定を受け付ける手順、

前記指定に従い、前記リードアクセスの前記発行先と、前記発行タイミン

グを決定する手順をコンピュータに実行させるためのプログラム。

[0183] 40. 複数のサーバがそれぞれ有する複数の分散テーブルからデータを読み出すリードアクセス処理のキー、および、前記キーに基づき読み出した結果を用いて実行する処理を定めるコールバック、を受け付ける受付手段と、

受け付けた前記リードアクセス処理を待機させ、該当するサーバに後で発行する待機リードアクセスとして前記キーと前記コールバックをアクセス待ちリストに登録する登録手段と、

前記アクセス待ちリストに登録された前記待機リードアクセスについて、所定の条件に従い、前記待機リードアクセスを発行する発行先のサーバおよび発行タイミングを決定する決定手段と、

前記決定手段が決定した前記発行先の前記サーバに、決定した前記発行タイミングで、前記待機リードアクセスを発行し、前記サーバから当該アクセス結果を受け取るリードアクセス発行手段と、

前記リードアクセス発行手段が受け取った前記アクセス結果を用いて、前記アクセス待ちリストに登録された該当する待機リードアクセスの前記コールバックを実行し、実行した前記コールバックの前記待機リードアクセスを前記アクセス待ちリストから削除するコールバック実行手段と、を備える情報処理装置。

41. 40. に記載の情報処理装置において、

前記登録手段は、前記サーバ毎、または前記キー毎に、前記待機リードアクセスを分類して前記アクセス待ちリストに登録し、

前記決定手段は、前記サーバ毎または前記キー毎に、前記待機リードアクセスの前記発行タイミングを決定する情報処理装置。

42. 40. または41. に記載の情報処理装置において、

前記決定手段は、各前記サーバの性能情報に基づいて求まる、前記キー1つあたりにかかる前記アクセス結果の取得時間に基づいて、前記待機リードアクセスの前記発行先となるサーバを選択する情報処理装置。

43. 40. 乃至42. いずれかに記載の情報処理装置において、

前記決定手段は、前記アクセス待ちリストに登録された前記待機リードアクセスの件数が一定数以上になったときを、前記発行タイミングであると決定する情報処理装置。

44. 40.乃至43. いずれかに記載の情報処理装置において、
前記決定手段は、

前記アクセス待ちリストのメモリ量が一定量以上になった場合、前記発行タイミングになったと判断し、

前記メモリ量が一定量以上になった時点で、前記アクセス待ちリストに登録された前記待機リードアクセスの件数が多いサーバを前記待機リードアクセスの前記発行先として選択する情報処理装置。

45. 40.乃至44. いずれかに記載の情報処理装置において、

前記決定手段は、一定時間以上、前記リードアクセス発行手段により前記待機リードアクセスが発行されていないサーバを前記待機リードアクセスの前記発行先として選択する情報処理装置。

46. 40.乃至45. いずれかに記載の情報処理装置において、

前記受付手段は、前記分散テーブルのキーに対して値を書き込むライトアクセス処理をさらに受け付け、

前記アクセス待ちリストを参照し、前記ライトアクセス処理と同じキーに対する前記待機リードアクセスが存在する場合、前記コールバック実行手段は、前記ライトアクセス処理の前記値で前記待機リードアクセスの前記コールバックを実行し、前記アクセス待ちリストから当該待機リードアクセスを削除する情報処理装置。

47. 46. に記載の情報処理装置において、

前記受付手段は、前記リードアクセス処理より後に受け付ける前記ライトアクセス処理を、先に処理することを禁止する禁止フラグを、前記リードアクセス処理の前記キーと前記コールバックとともに受け付け、

前記登録手段は、受け付けた前記禁止フラグを前記待機リードアクセスの前記キーおよび前記コールバックとともに前記アクセス待ちリストに登録し

、

さらに、前記登録手段は、前記アクセス待ちリストを参照し、受け付けた前記ライトアクセス処理と同じキーに対する前記待機リードアクセスの前記禁止フラグが登録されている場合、受け付けた前記ライトアクセス処理を待機させ、該当するサーバに後で発行する待機ライトアクセスとして前記キーと前記値を前記アクセス待ちリストに登録する情報処理装置。

48. 42. に記載の情報処理装置において、

前記決定手段は、前記キー1つあたりにかかる前記アクセス結果の取得時間を前記サーバ毎に算出し、最小値が閾値以下のサーバを前記リードアクセスの前記発行先として選択する情報処理装置。

49. 48. に記載の情報処理装置において、

前記決定手段は、前記登録手段が、前記アクセス待ちリストに前記待機リードアクセスを登録する度に、前記キー1つあたりにかかる前記アクセス結果の取得時間を算出し、算出される度に、前記発行先の選択を行う情報処理装置。

50. 48. に記載の情報処理装置において、

前記決定手段は、前記登録手段による前記アクセス待ちリストへの前記待機リードアクセスの登録が所定回数毎に、前記キー1つあたりにかかる前記アクセス結果の取得時間を算出し、算出される度に、前記発行先の選択を行う情報処理装置。

51. 48. に記載の情報処理装置において、

前記決定手段は、一定時間毎に、前記キー1つあたりにかかる前記アクセス結果の取得時間を算出し、算出される度に、前記発行先の選択を行う情報処理装置。

52. 40. 乃至51. いずれかに記載の情報処理装置において、

前記決定手段は、前記リードアクセスの前記発行先と、前記発行タイミングの指定を受け付け、前記指定に従い、前記リードアクセスの前記発行先と、前記発行タイミングを決定する情報処理装置。

[0184] 53. 40. 乃至52. いずれかに記載の情報処理装置と、

前記情報処理装置がアクセスする複数の分散テーブルをそれぞれ有する複数のサーバと、を備える分散システム。

[0185] この出願は、2012年8月31日に出願された日本特許出願特願2012-192453を基礎とする優先権を主張し、その開示の全てをここに取り込む。

請求の範囲

[請求項1]

複数のサーバがそれぞれ有する複数の分散テーブルからデータを読み出すリードアクセス処理のキー、および、前記キーに基づき読み出した結果を用いて実行する処理を定めるコールバック、を受け付ける受付手段と、

受け付けた前記リードアクセス処理を待機させ、該当するサーバに後で発行する待機リードアクセスとして前記キーと前記コールバックをアクセス待ちリストに登録する登録手段と、

前記アクセス待ちリストに登録された前記待機リードアクセスについて、所定の条件に従い、前記待機リードアクセスを発行する発行先のサーバおよび発行タイミングを決定する決定手段と、

前記決定手段が決定した前記発行先の前記サーバに、決定した前記発行タイミングで、前記待機リードアクセスを発行し、前記サーバから当該アクセス結果を受け取るリードアクセス発行手段と、

前記リードアクセス発行手段が受け取った前記アクセス結果を用いて、前記アクセス待ちリストに登録された該当する待機リードアクセスの前記コールバックを実行し、実行した前記コールバックの前記待機リードアクセスを前記アクセス待ちリストから削除するコールバック実行手段と、を備える分散システム。

[請求項2]

請求項1に記載の分散システムにおいて、

前記登録手段は、前記サーバ毎、または前記キー毎に、前記待機リードアクセスを分類して前記アクセス待ちリストに登録し、

前記決定手段は、前記サーバ毎または前記キー毎に、前記待機リードアクセスの前記発行タイミングを決定する分散システム。

[請求項3]

請求項1または2に記載の分散システムにおいて、

前記決定手段は、前記アクセス待ちリストに登録された前記待機リードアクセスの件数が一定数以上になったときを、前記発行タイミングであると決定する分散システム。

- [請求項4] 請求項1乃至3いずれかに記載の分散システムにおいて、
前記決定手段は、
前記アクセス待ちリストのメモリ量が一定量以上になった場合、
前記発行タイミングになったと判断し、
前記メモリ量が一定量以上になった時点で、前記アクセス待ちリストに登録された前記待機リードアクセスの件数が多いサーバを前記待機リードアクセスの前記発行先として選択する分散システム。
- [請求項5] 請求項1乃至4いずれかに記載の分散システムにおいて、
前記決定手段は、各前記サーバの性能情報に基づいて求まる、前記キー1つあたりにかかる前記アクセス結果の取得時間に基づいて、前記待機リードアクセスの前記発行先となるサーバを選択する分散システム。
- [請求項6] 請求項1乃至5いずれかに記載の分散システムにおいて、
前記決定手段は、一定時間以上、前記リードアクセス発行手段により前記待機リードアクセスが発行されていないサーバを前記待機リードアクセスの前記発行先として選択する分散システム。
- [請求項7] 請求項1乃至6いずれかに記載の分散システムにおいて、
前記受付手段は、前記分散テーブルのキーに対して値を書き込むライトアクセス処理をさらに受け付け、
前記アクセス待ちリストを参照し、前記ライトアクセス処理と同じキーに対する前記待機リードアクセスが存在する場合、前記コールバック実行手段は、前記ライトアクセス処理の前記値で前記待機リードアクセスの前記コールバックを実行し、前記アクセス待ちリストから当該待機リードアクセスを削除する分散システム。
- [請求項8] 請求項7に記載の分散システムにおいて、
前記受付手段は、前記リードアクセス処理より後に受け付ける前記ライトアクセス処理を、先に処理することを禁止する禁止フラグを、前記リードアクセス処理の前記キーと前記コールバックとともに受け

付け、

前記登録手段は、受け付けた前記禁止フラグを前記待機リードアクセスの前記キーおよび前記コールバックとともに前記アクセス待ちリストに登録し、

さらに、前記登録手段は、前記アクセス待ちリストを参照し、前記ライトアクセス処理と同じキーに対する前記待機リードアクセスの前記禁止フラグが登録されている場合、前記ライトアクセス処理を待機させ、該当するサーバに後で発行する待機ライトアクセスとして前記キーと前記値を前記アクセス待ちリストに登録する分散システム。

[請求項9]

情報処理装置が、

複数のサーバがそれぞれ有する複数の分散テーブルからデータを読み出すリードアクセス処理のキー、および、前記キーに基づき読み出した結果を用いて実行する処理を定めるコールバック、を受け付け、

受け付けた前記リードアクセス処理を待機させ、該当するサーバに後で発行する待機リードアクセスとして前記キーと前記コールバックをアクセス待ちリストに登録し、

前記アクセス待ちリストに登録された前記リードアクセスについて、所定の条件に従い、前記待機リードアクセスを発行するサーバおよび発行タイミングを決定し、

決定した前記サーバに、決定した前記発行タイミングで、前記待機リードアクセスを発行し、前記サーバから当該アクセス結果を受け取り、

受け取った前記アクセス結果を用いて、前記アクセス待ちリストに登録された該当する待機リードアクセスの前記コールバックを実行し、実行した前記コールバックの前記待機リードアクセスを前記アクセス待ちリストから削除する情報処理装置のデータ処理方法。

[請求項10]

情報処理装置を実現するコンピュータに、

複数のサーバがそれぞれ有する複数の分散テーブルからデータを読

み出すリードアクセス処理のキー、および、前記キーに基づき読み出した結果を用いて実行する処理を定めるコールバック、を受け付ける手順、

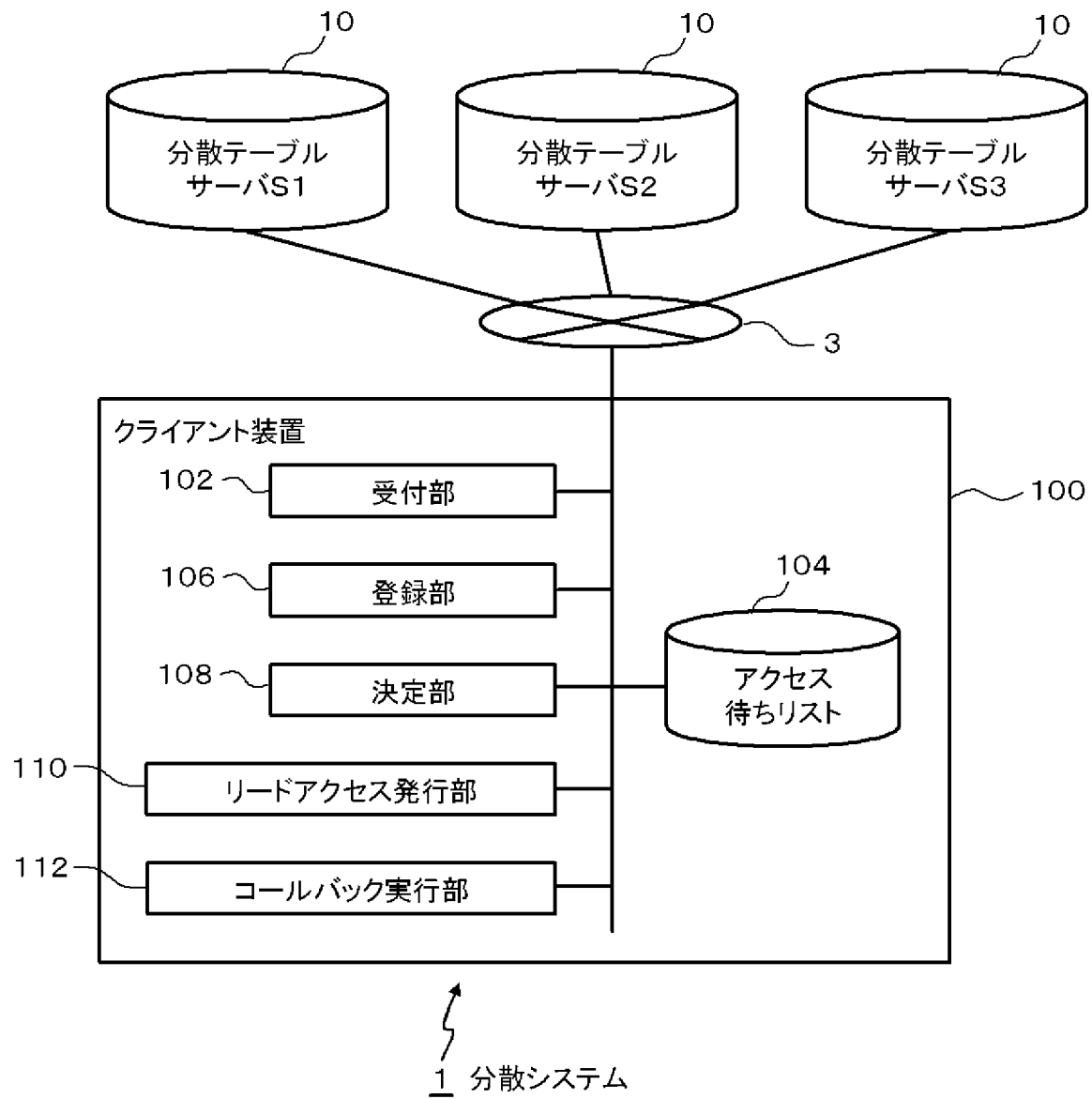
受け付けた前記リードアクセス処理を待機させ、該当するサーバに後で発行する待機リードアクセスとして前記キーと前記コールバックをアクセス待ちリストに登録する手順、

前記アクセス待ちリストに登録された前記待機リードアクセスについて、所定の条件に従い、前記待機リードアクセスを発行するサーバおよび発行タイミングを決定する手順、

決定した前記サーバに、決定した前記発行タイミングで、前記待機リードアクセスを発行し、前記サーバから当該アクセス結果を受け取る手順、

受け取った前記アクセス結果を用いて、前記アクセス待ちリストに登録された該当する待機リードアクセスの前記コールバックを実行し、実行した前記コールバックの前記待機リードアクセスを前記アクセス待ちリストから削除する手順、を実行させるためのプログラム。

[図1]



[図2]

(a)

104

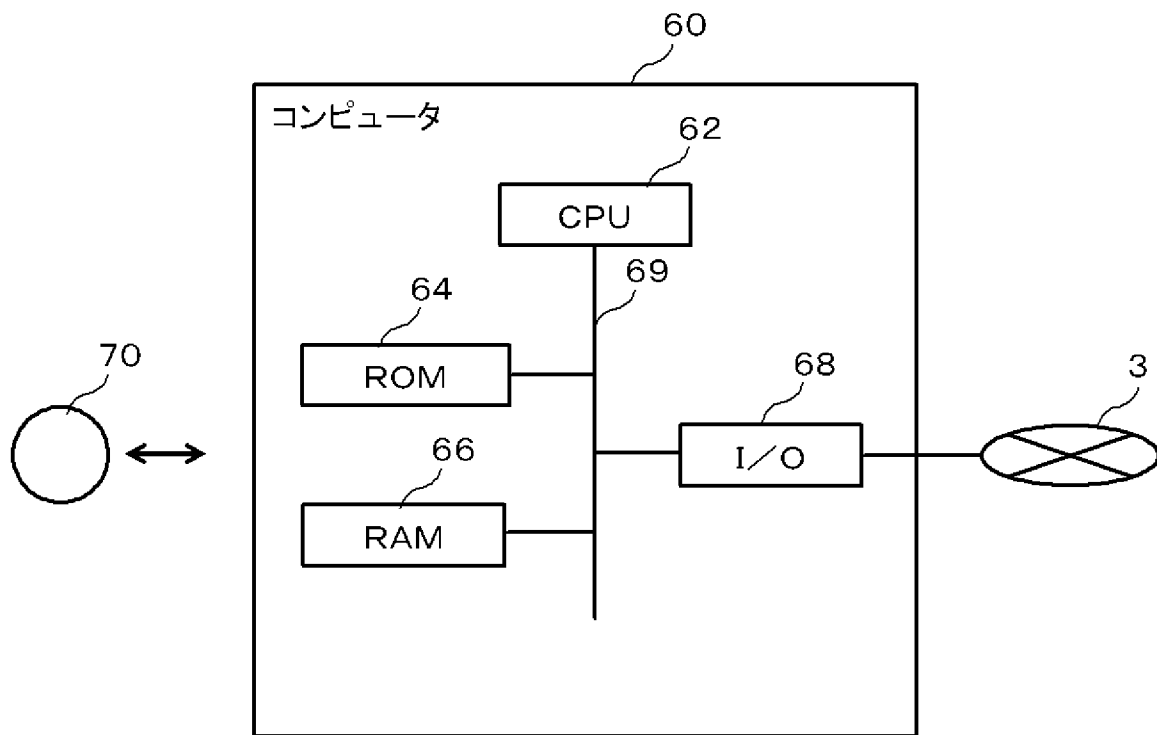
担当 サーバ	Key+Callbackリスト
S1	{Key1,Callback1}, {Key2,Callback2}, ...
S2	{Key3,Callback3} , ...
S3	{Key4,Callback4}, ...
⋮	⋮

(b)

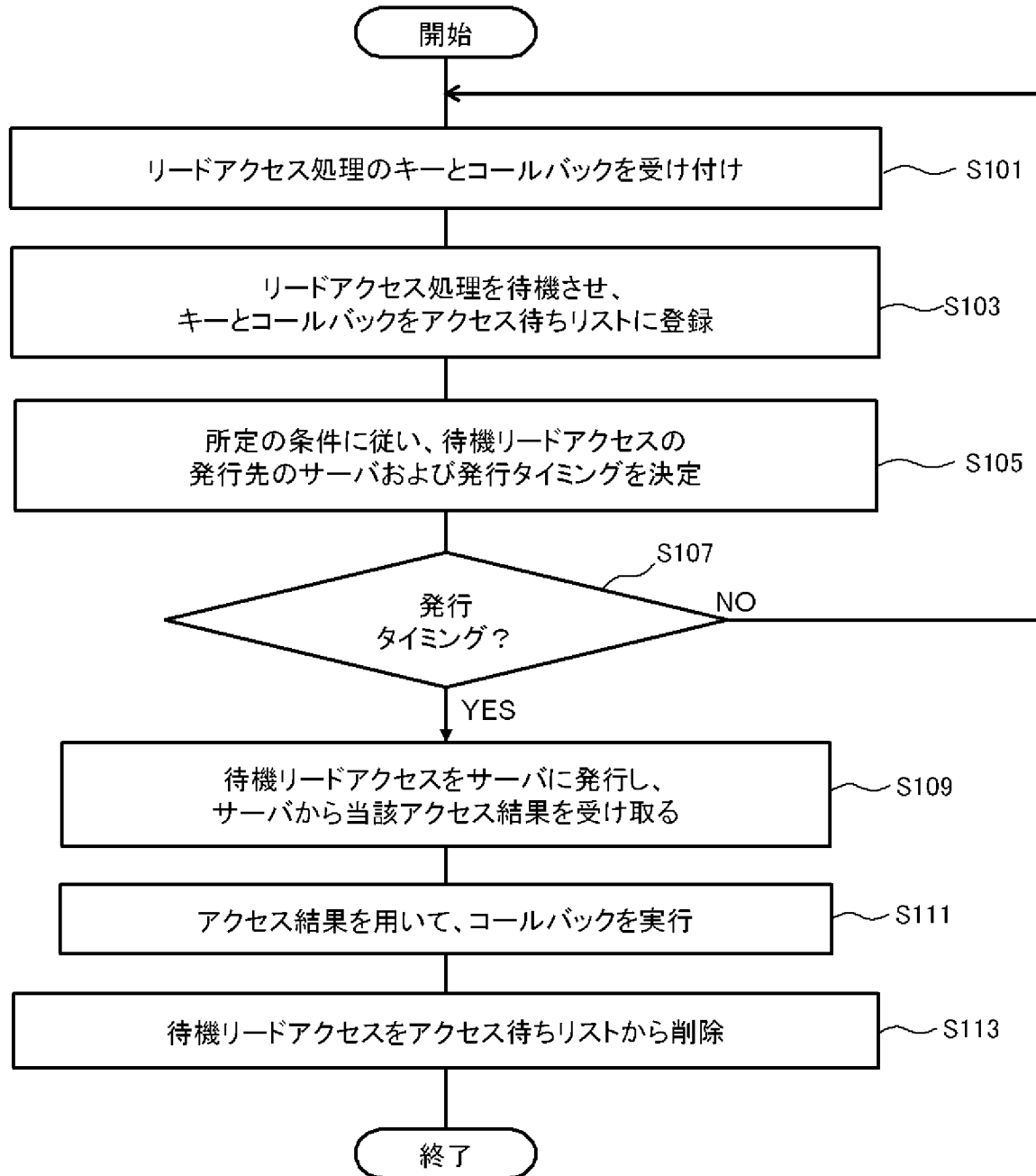
104

キー	Key+Callbackリスト
Key1	{Key1,Callback1}, ...
Key2	{Key2,Callback2}, ...
Key3	{Key3,Callback3} , ...
Key4	{Key4,Callback4}, ...
⋮	⋮

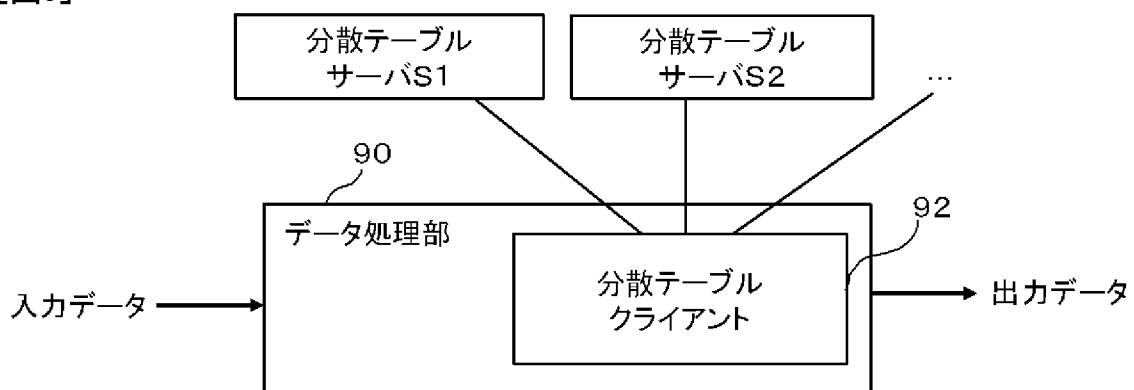
[図3]



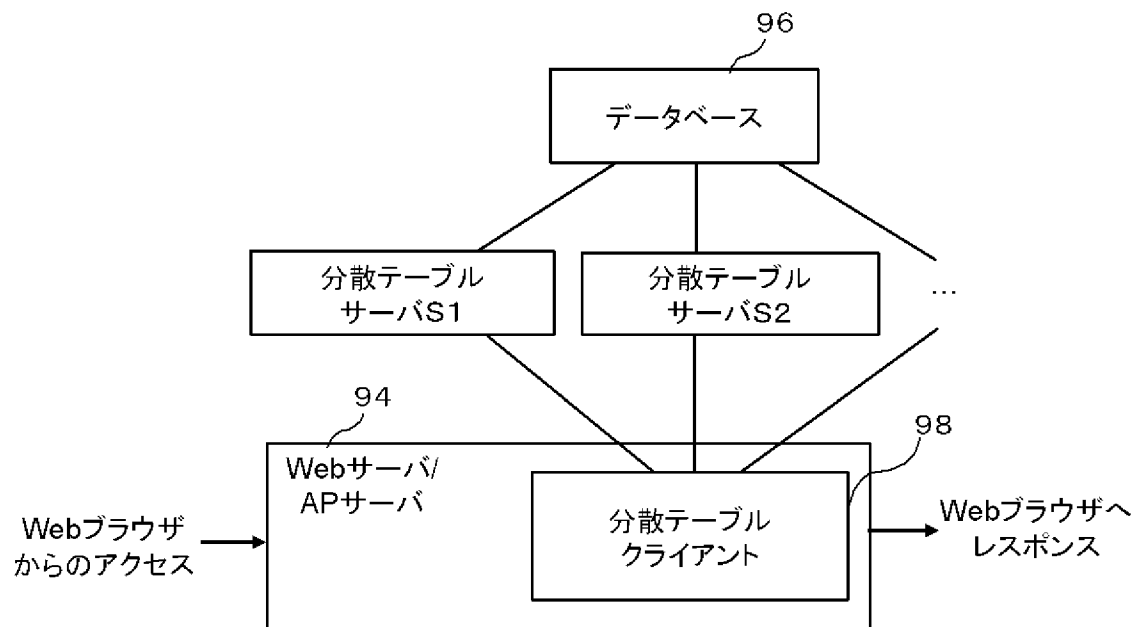
[図4]



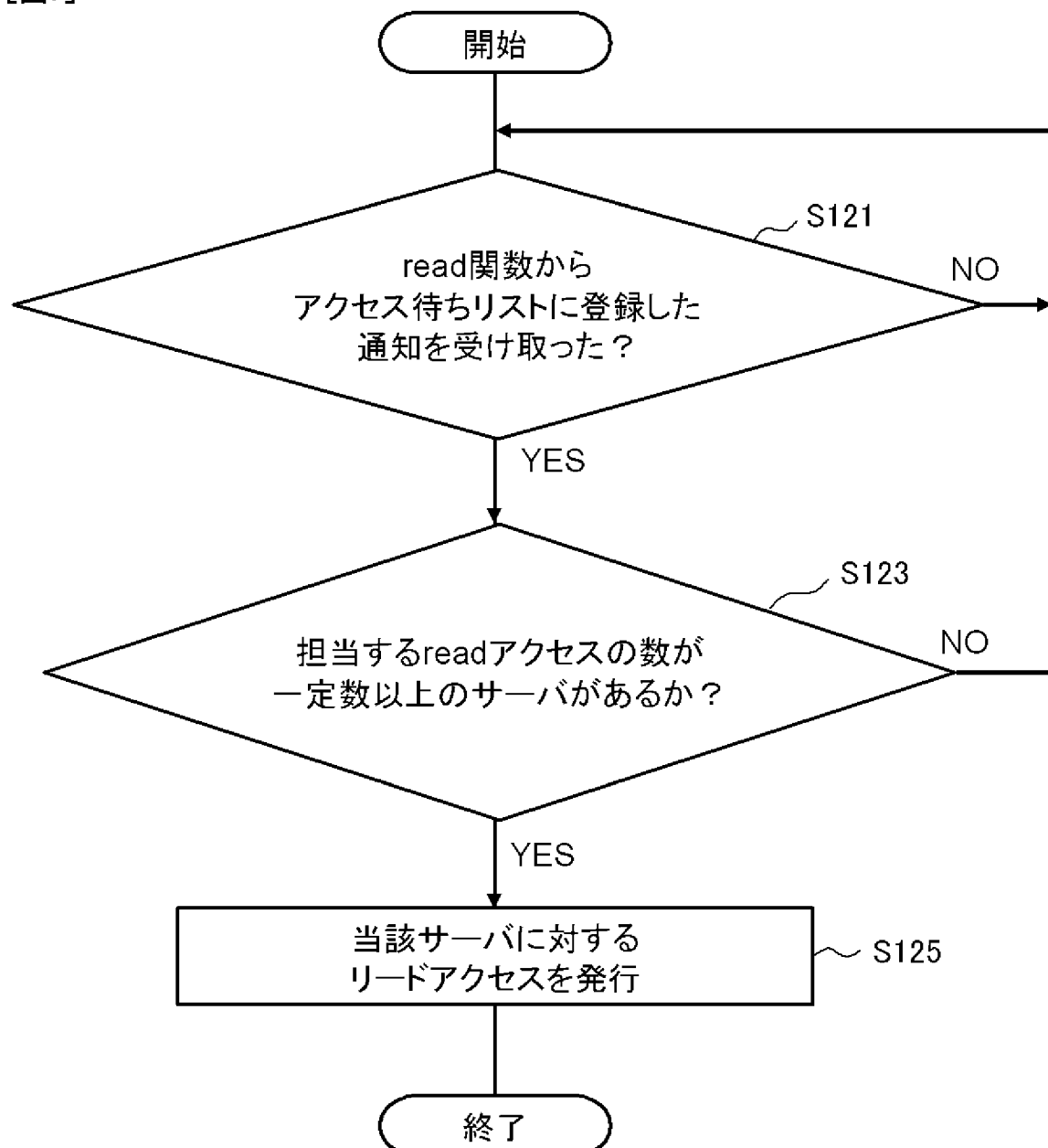
[図5]



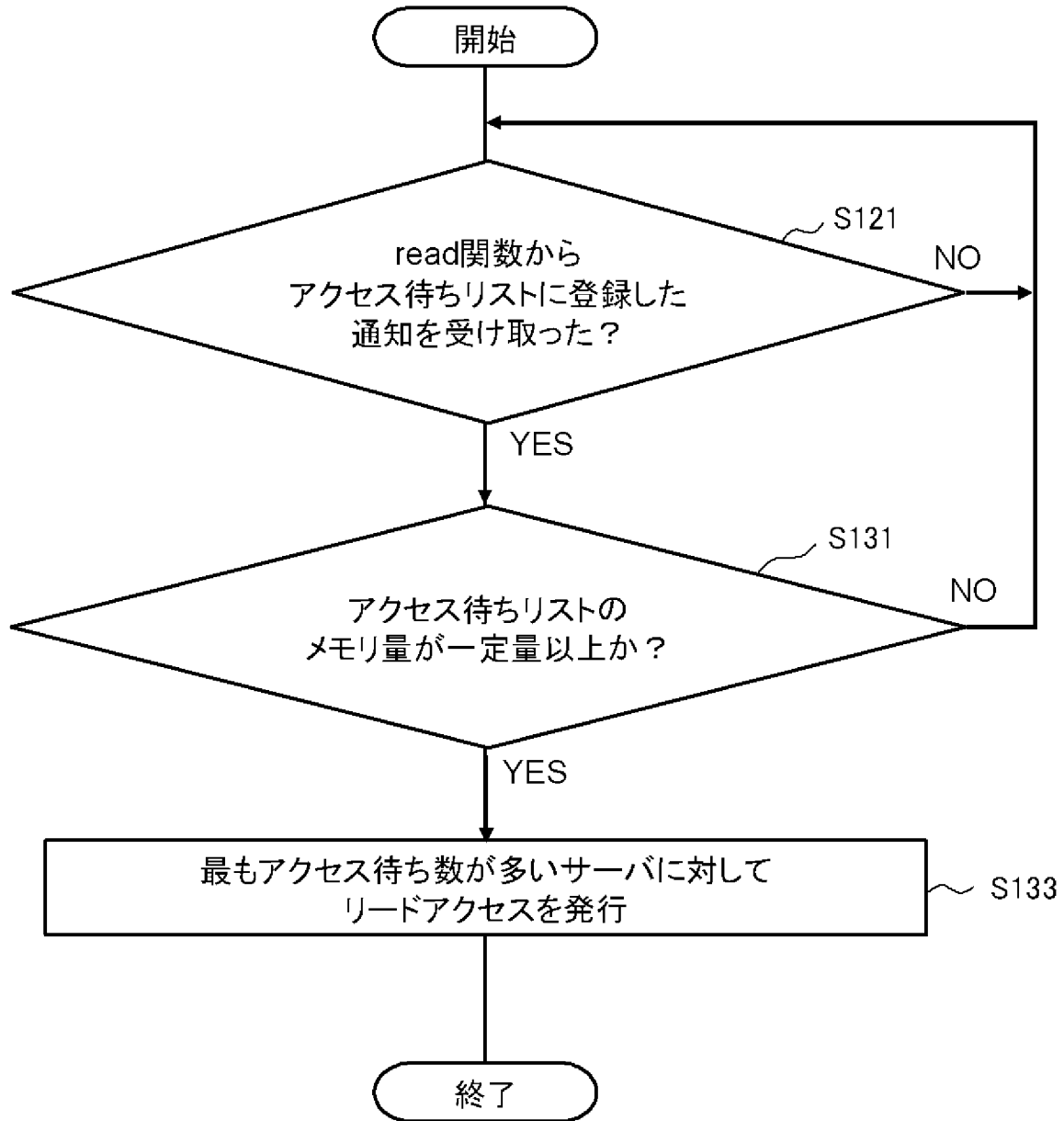
[図6]



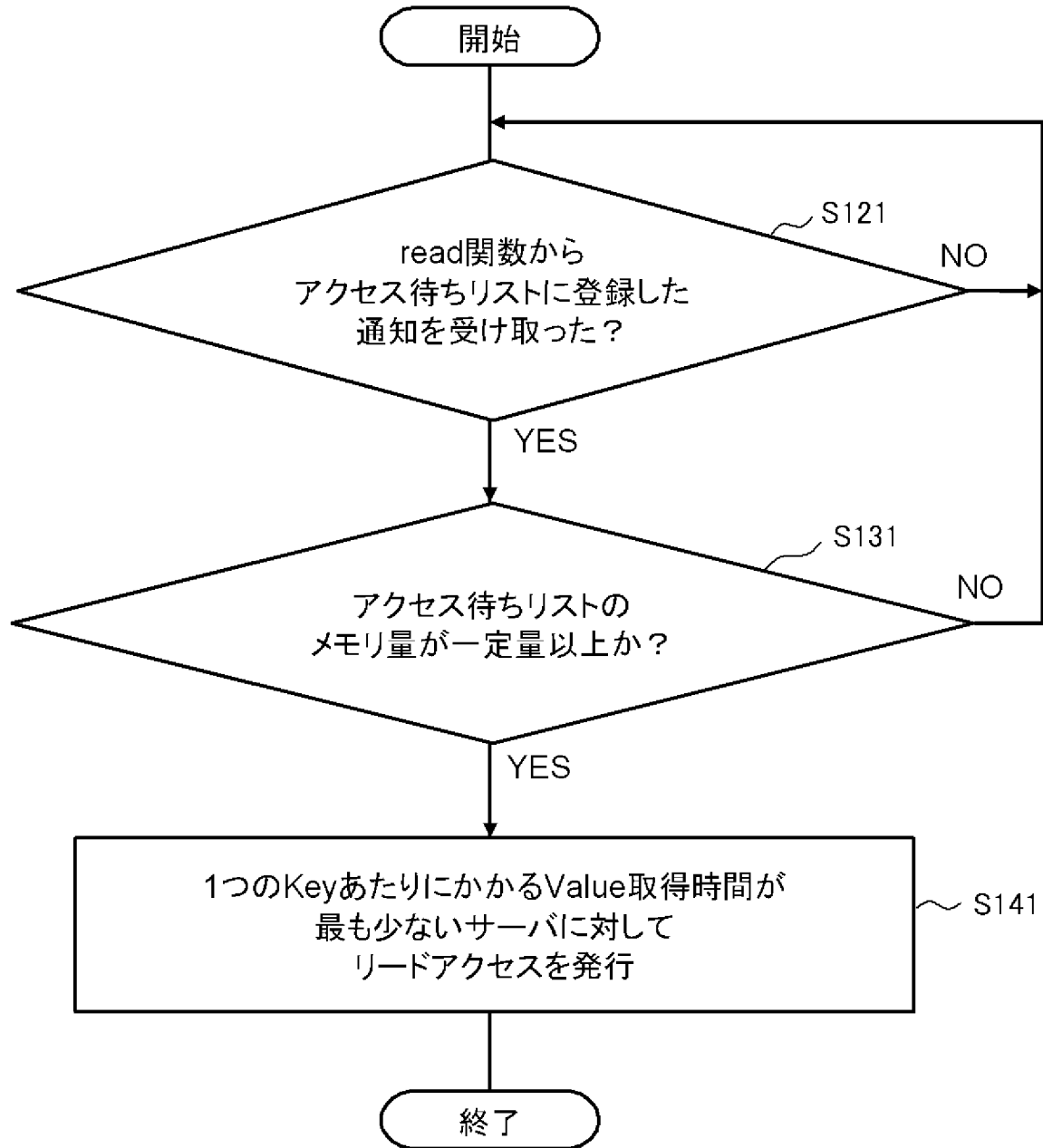
[図7]



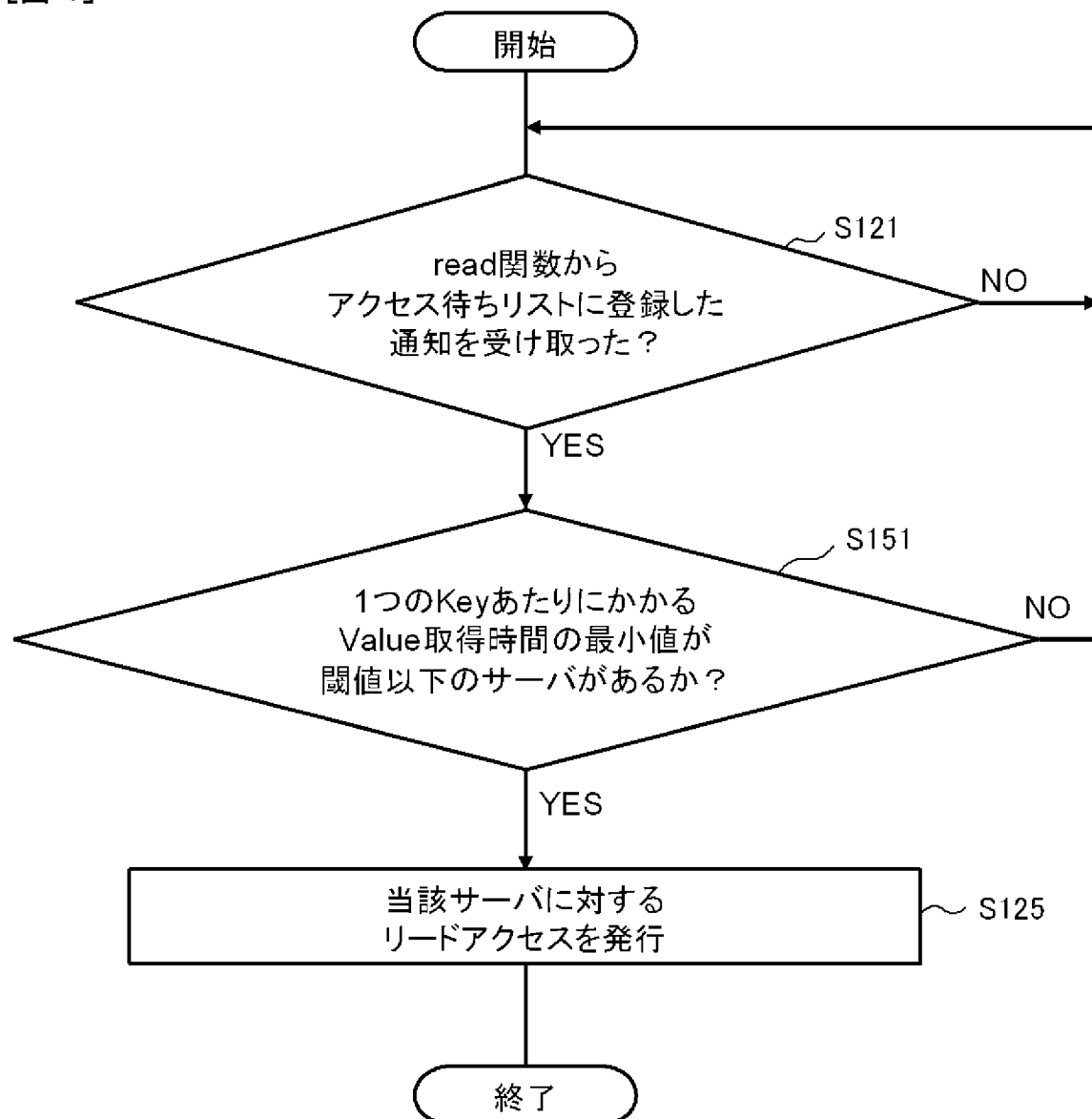
[図8]



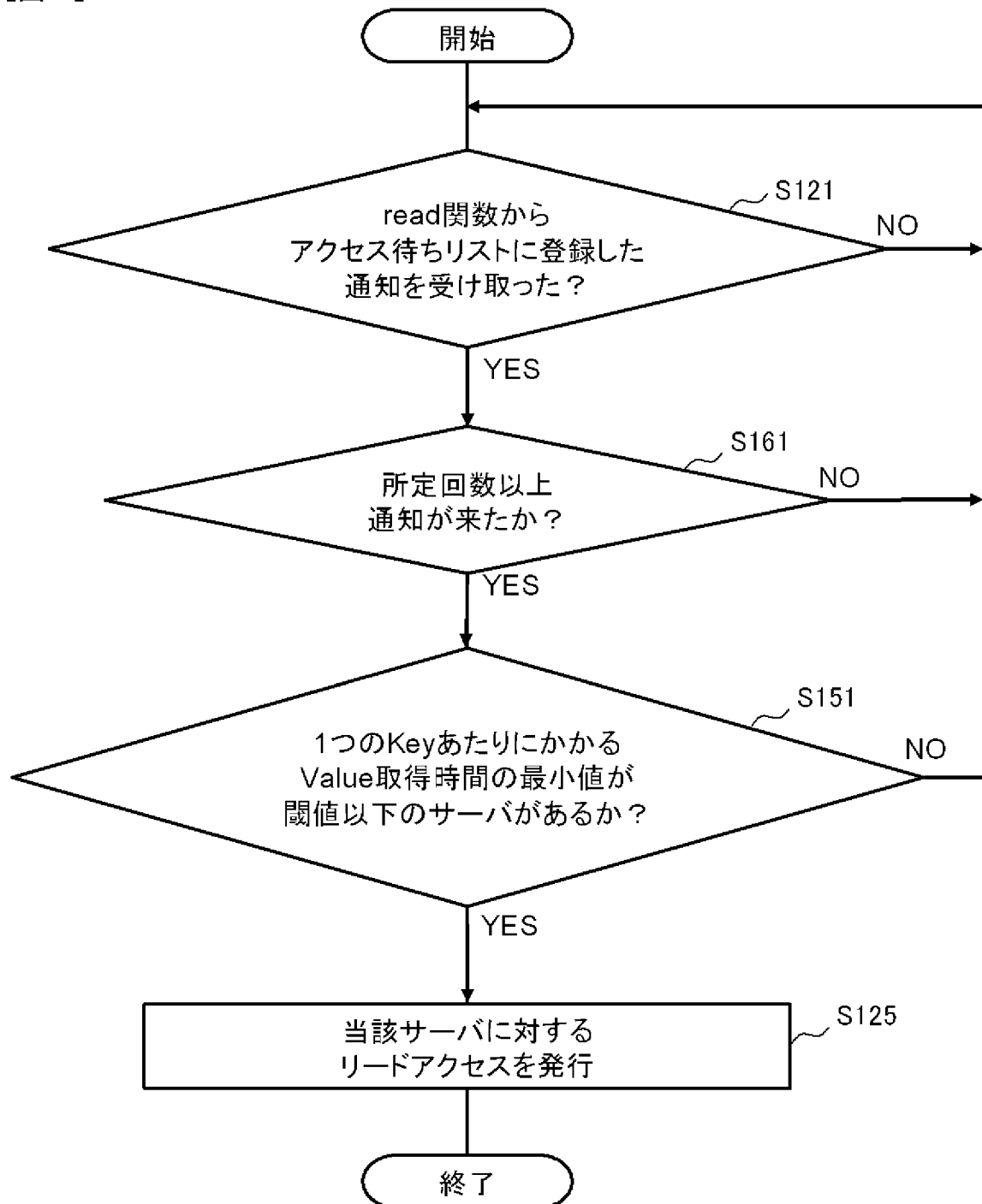
[図9]



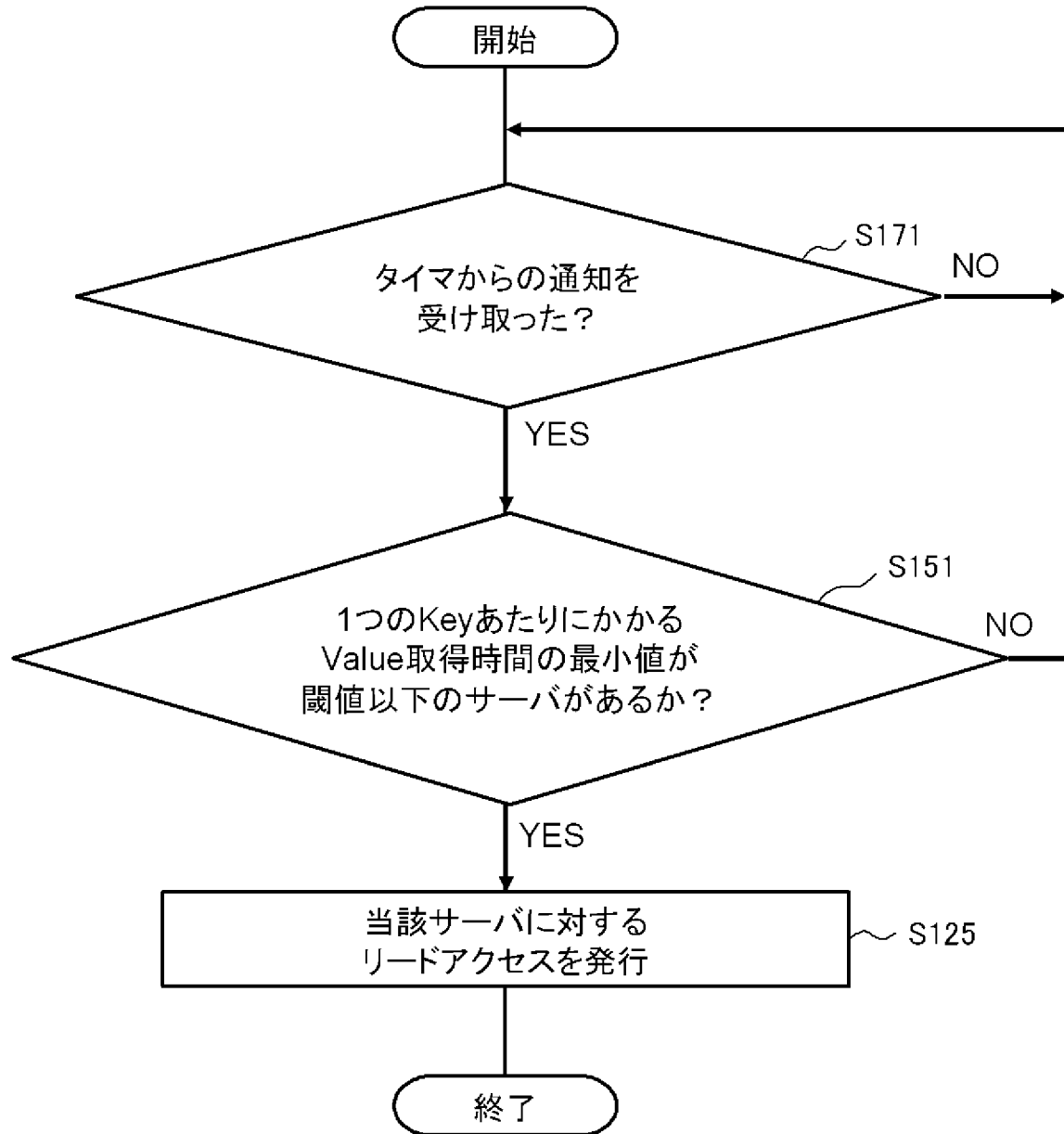
[図10]



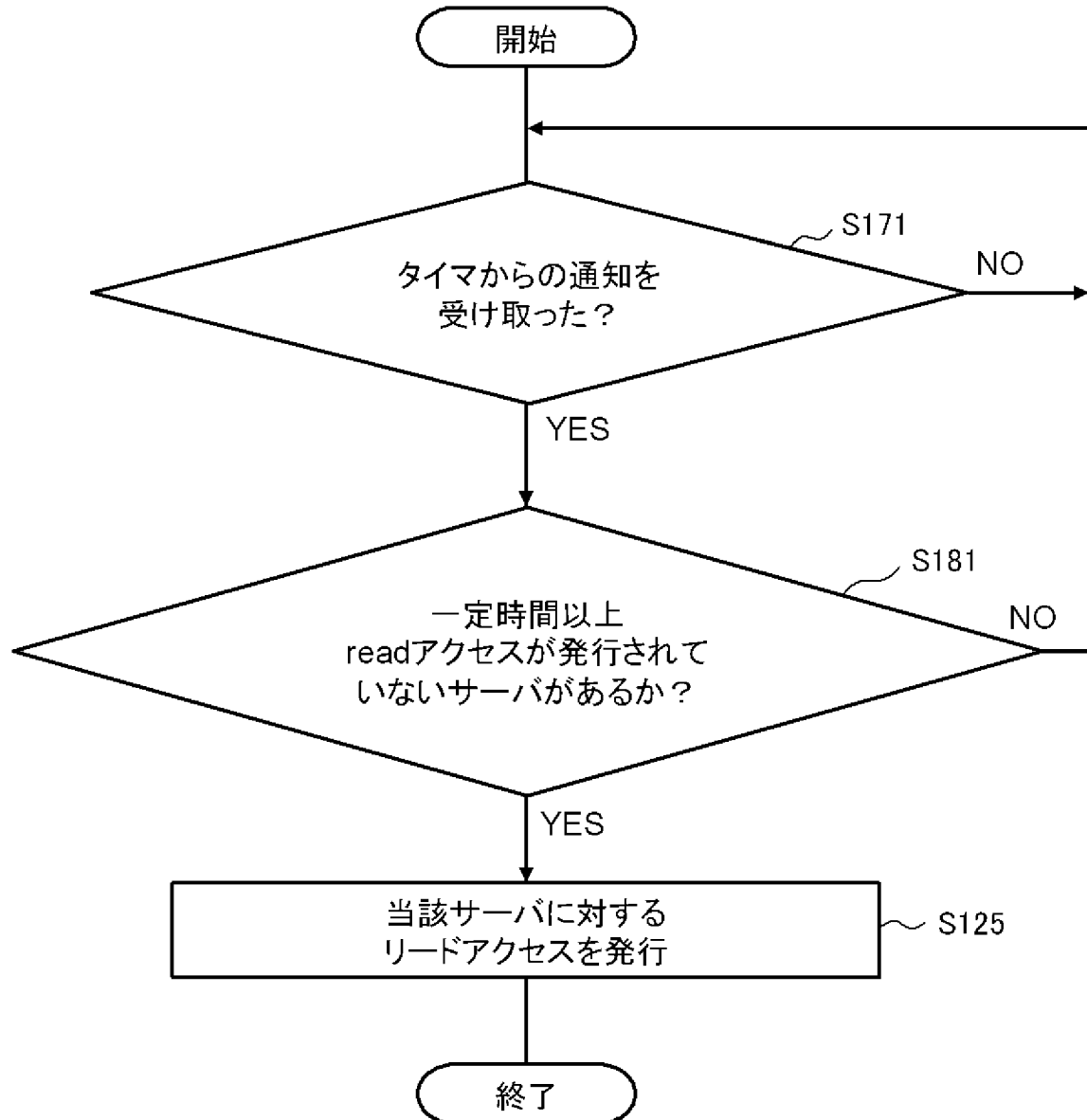
[図11]



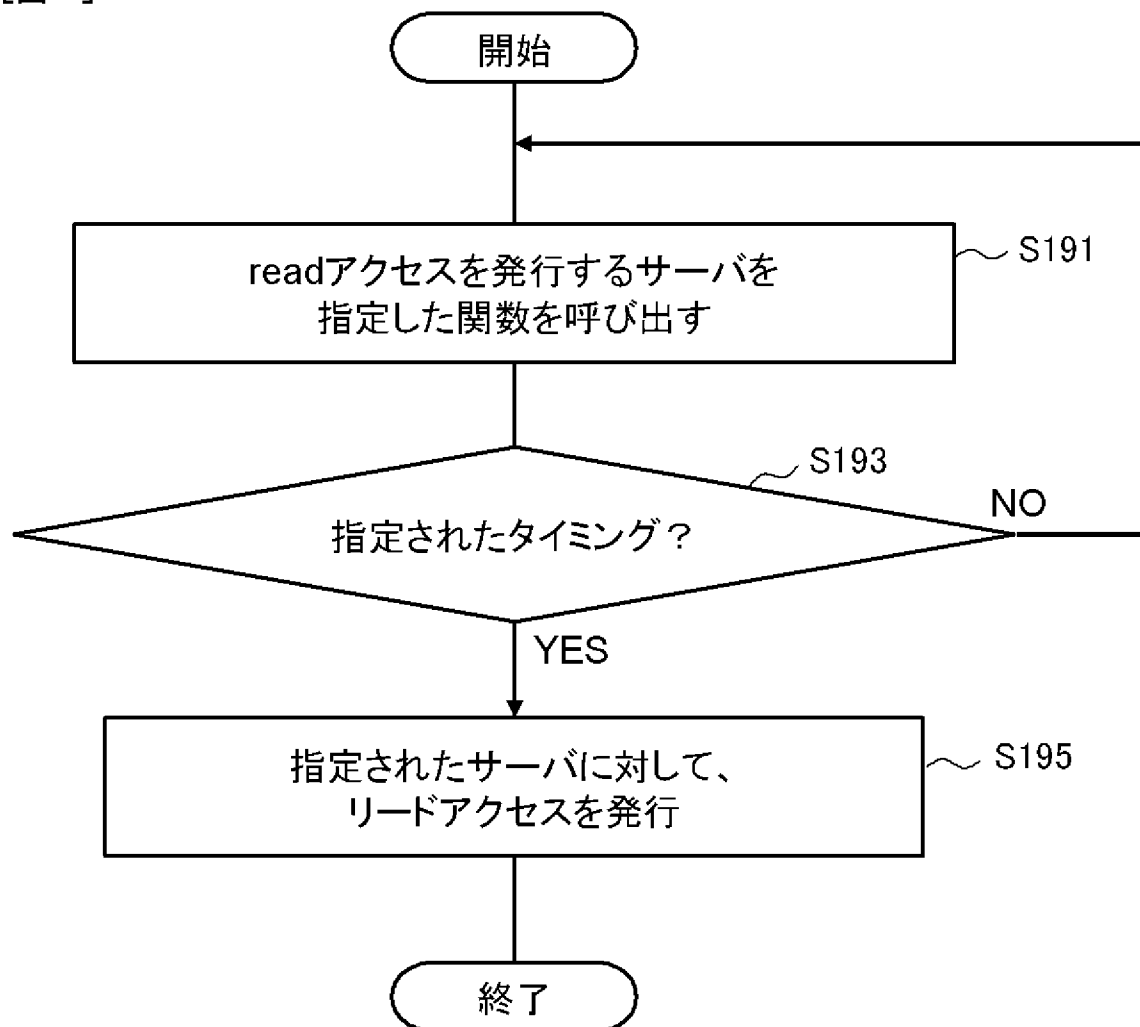
[図12]



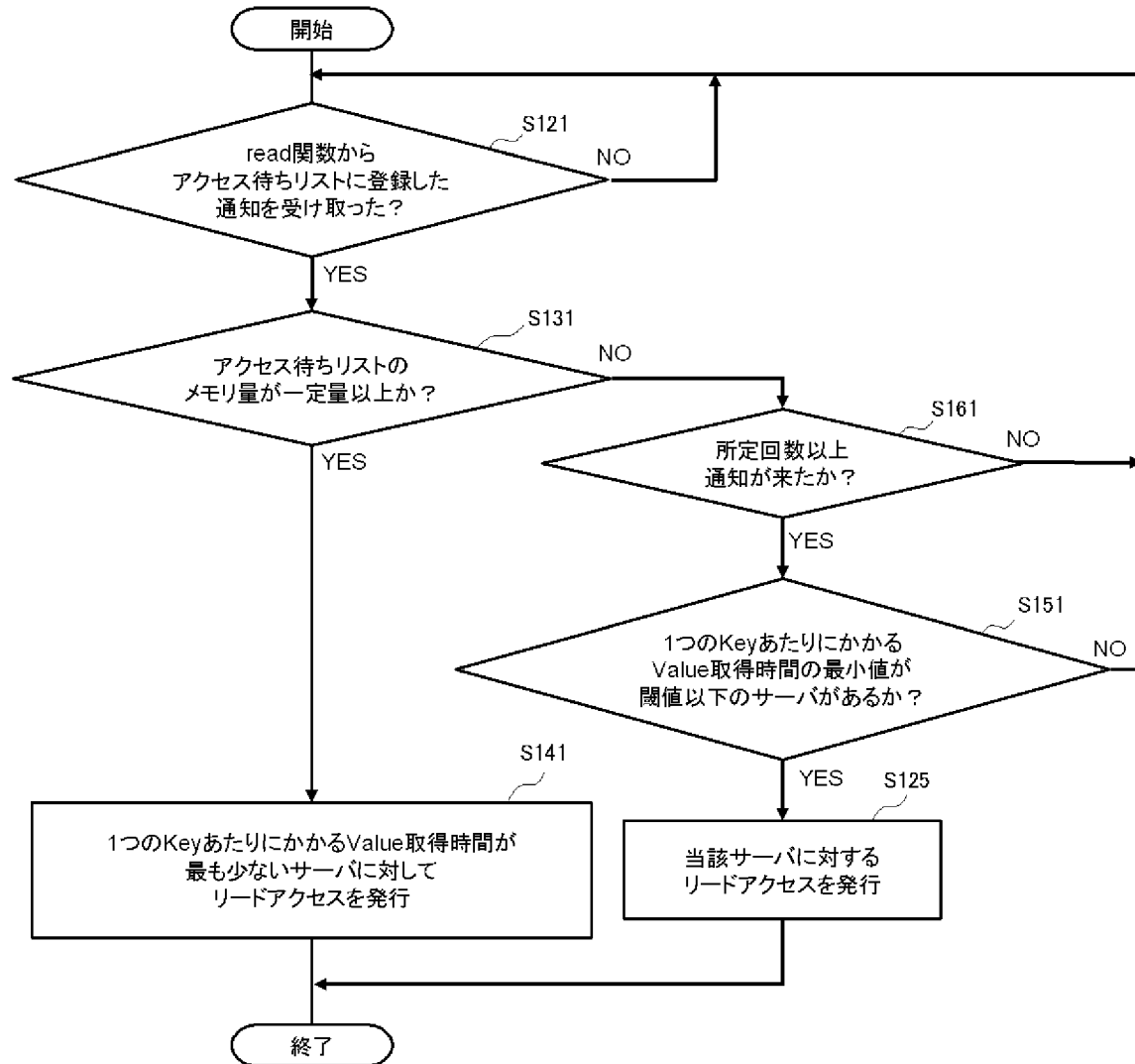
[図13]



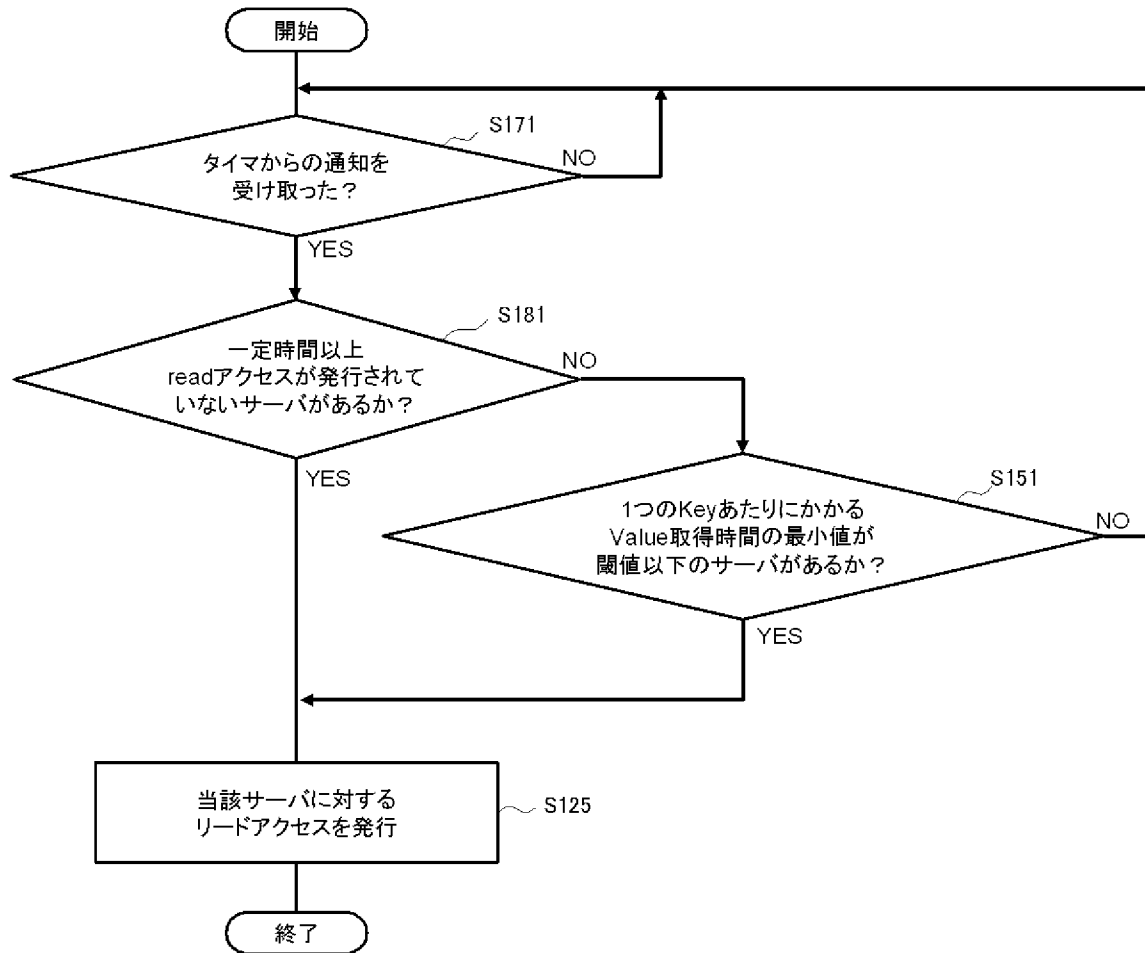
[図14]



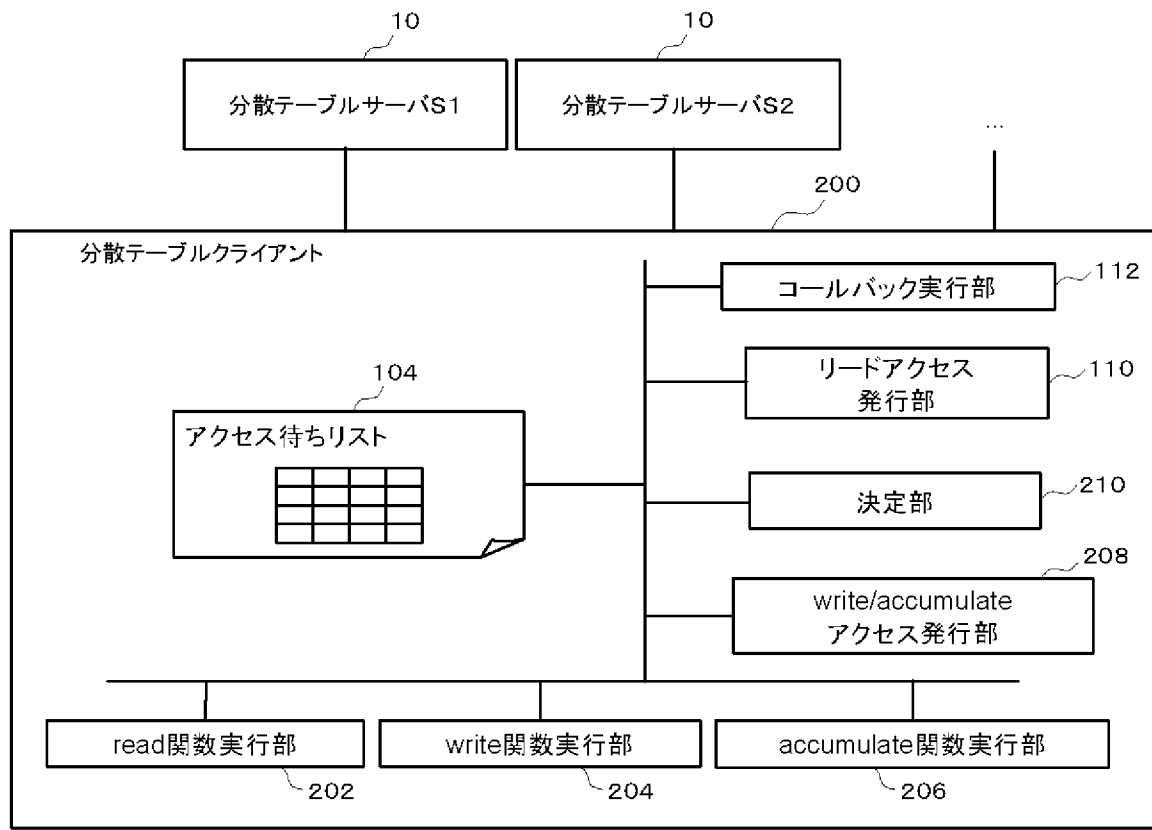
[図15]



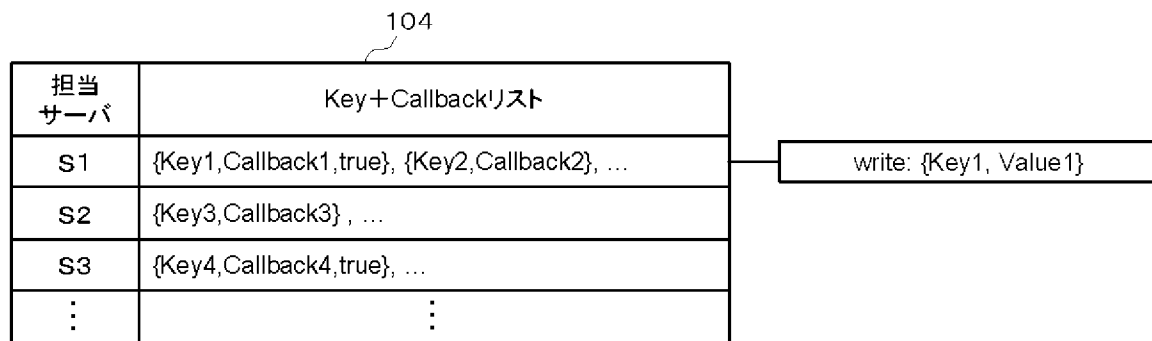
[図16]



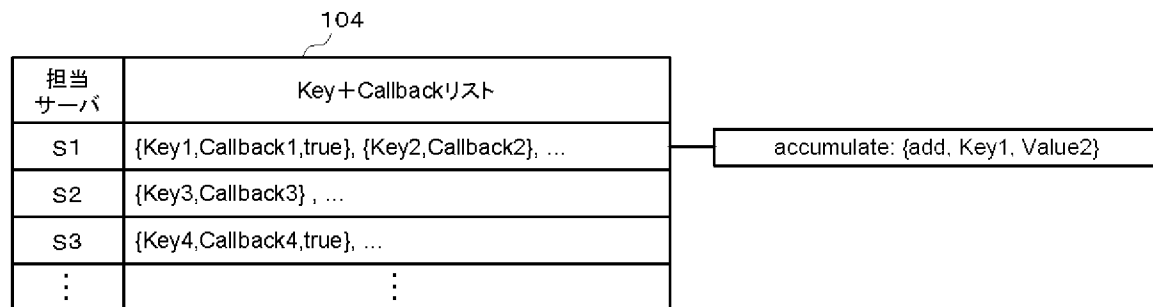
[図17]



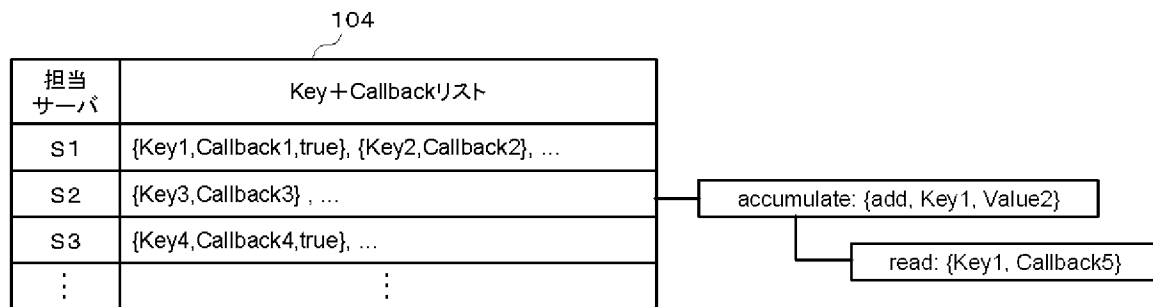
[図18]



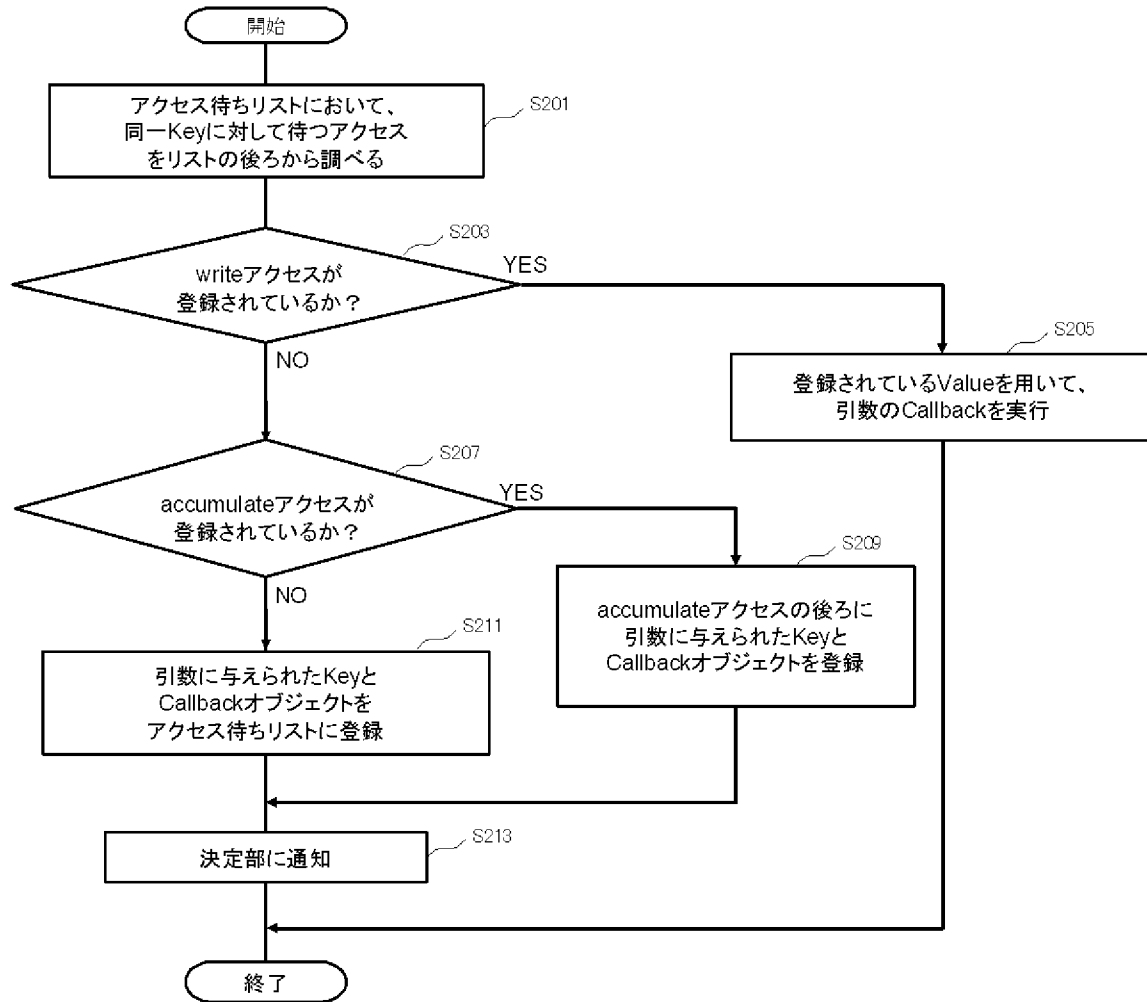
[図19]



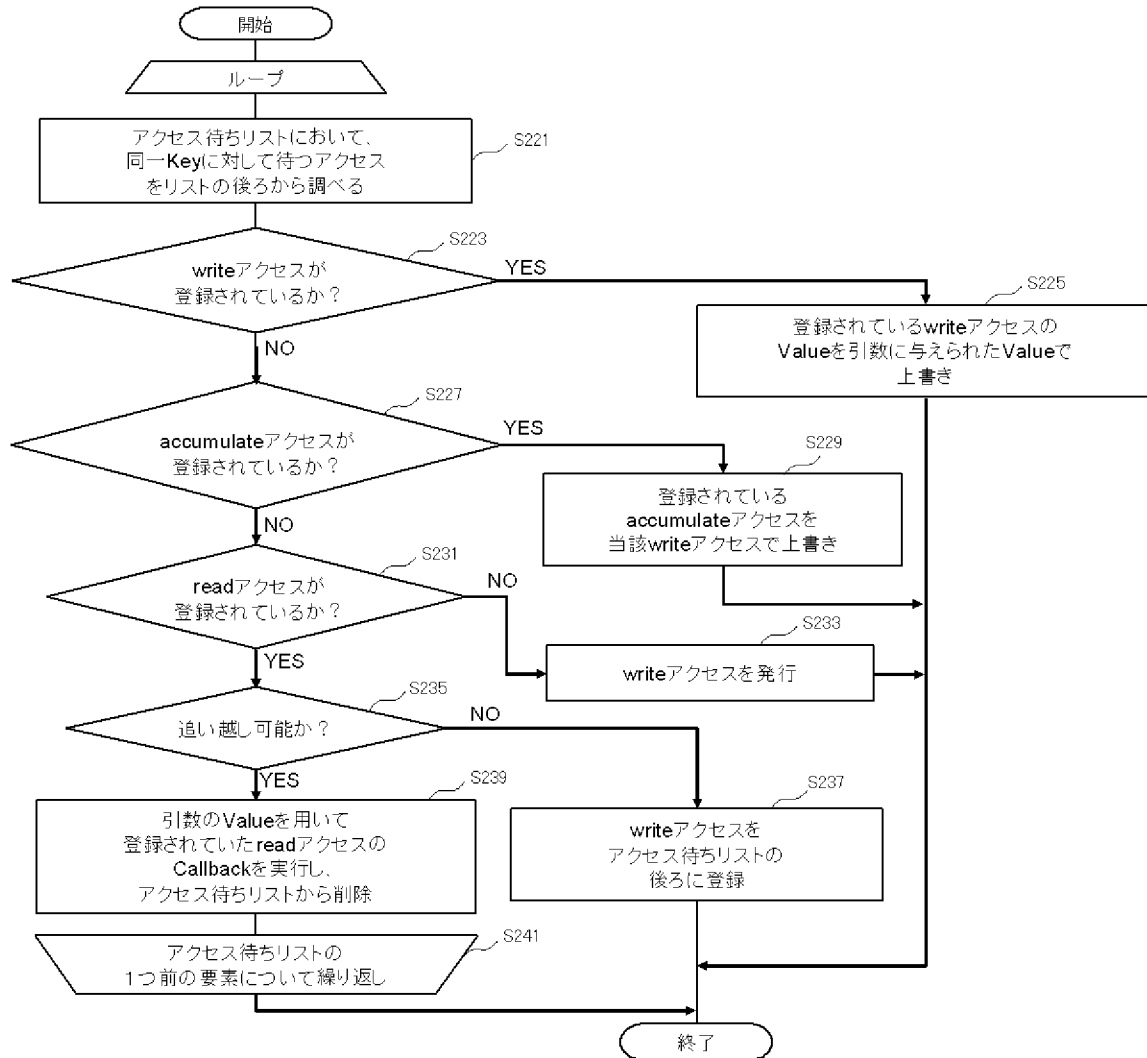
[図20]



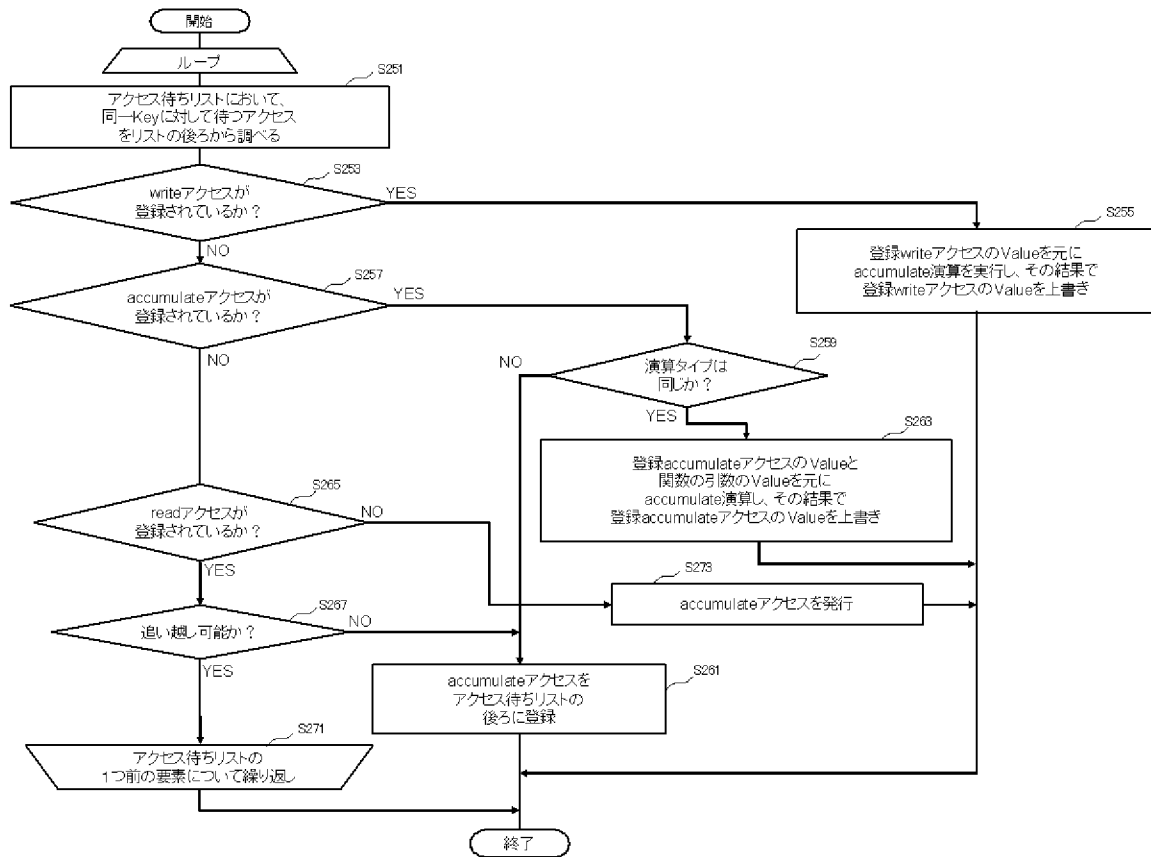
[図21]



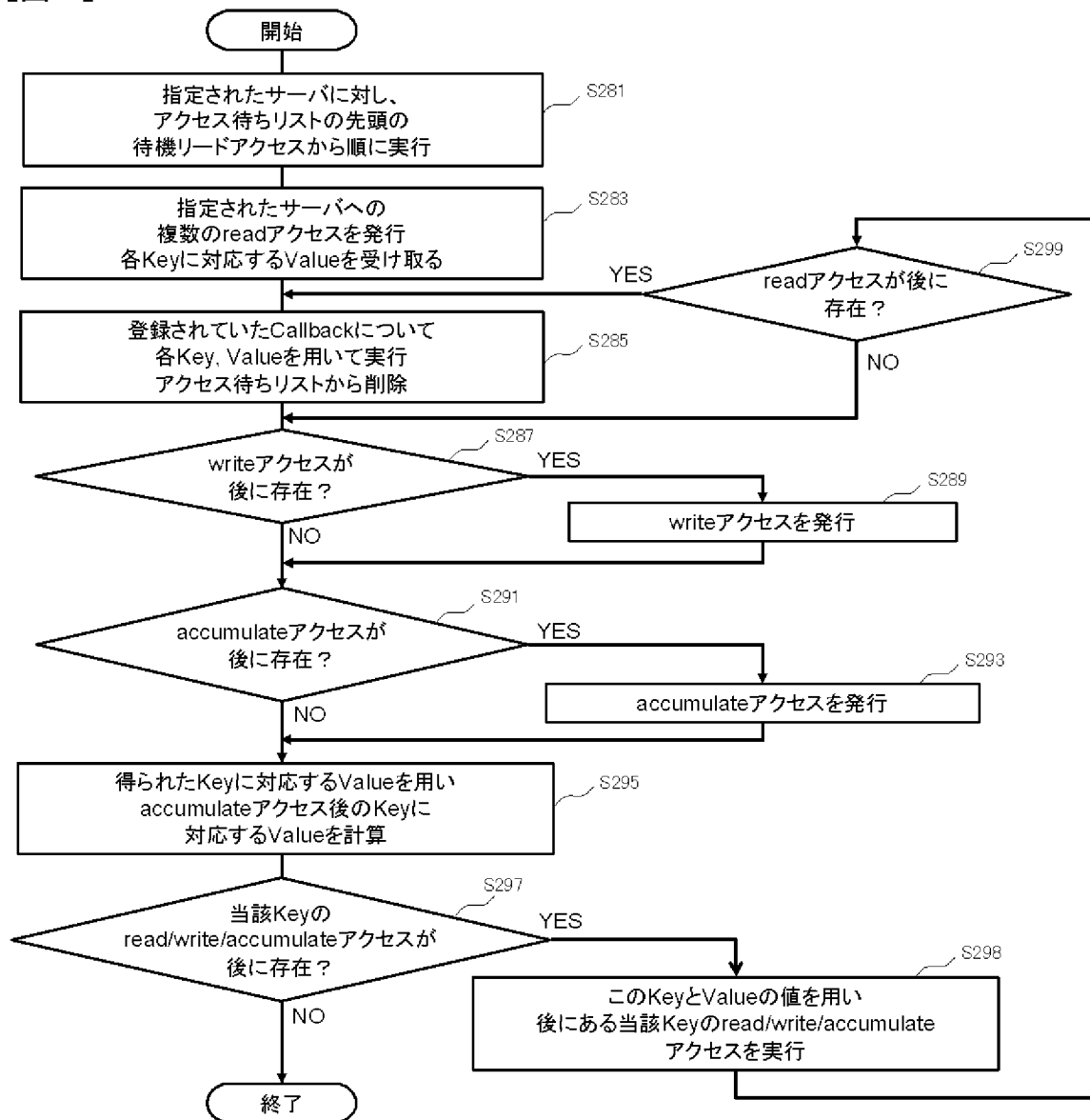
[図22]



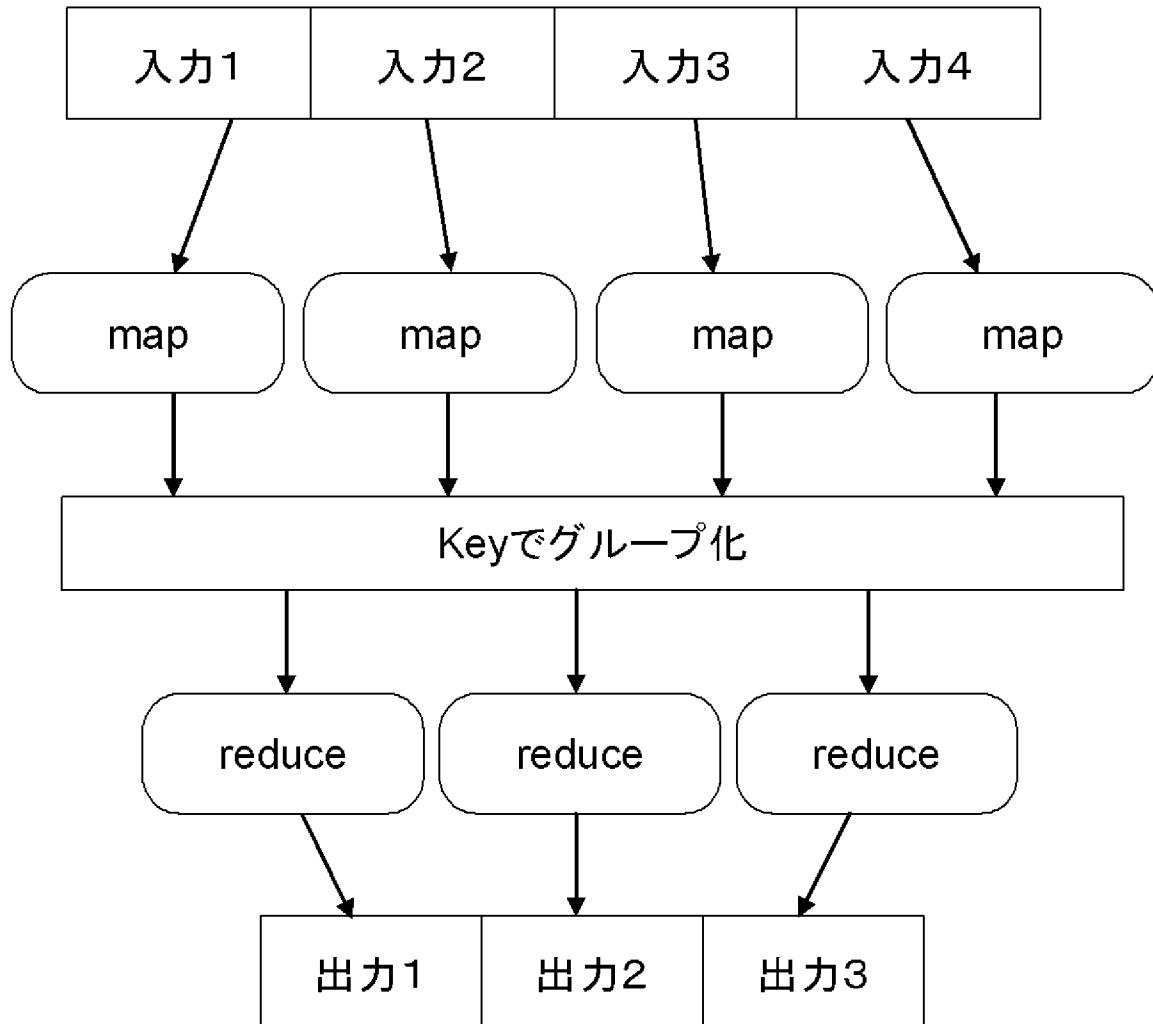
[図23]



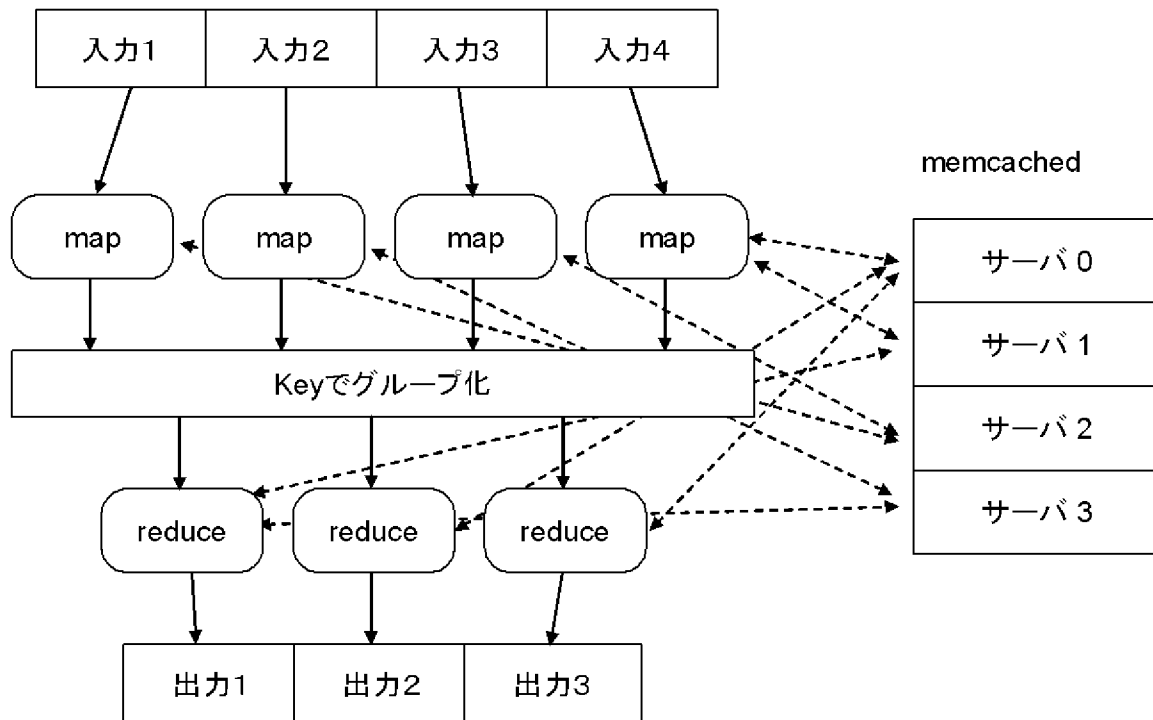
[図24]



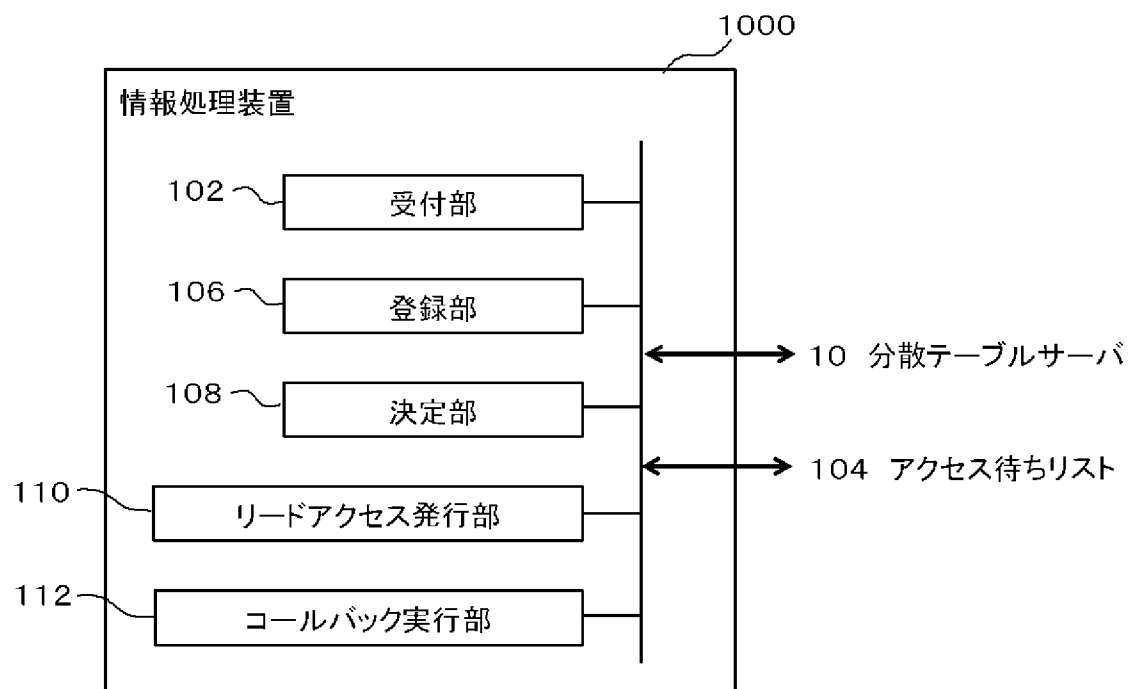
[図25]



[図26]



[図27]



INTERNATIONAL SEARCH REPORT

International application No.

PCT/JP2013/073328

A. CLASSIFICATION OF SUBJECT MATTER

G06F12/00(2006.01) i, G06F17/30(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, G06F17/30

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	1996-2013
Kokai Jitsuyo Shinan Koho	1971-2013	Toroku Jitsuyo Shinan Koho	1994-2013

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	WO 2012/108175 A1 (NEC Corp.), 16 August 2012 (16.08.2012), paragraphs [0013] to [0030] (Family: none)	1-10
A	JP 2011-233033 A (Fujitsu Ltd.), 17 November 2011 (17.11.2011), paragraphs [0013] to [0015], [0028], [0055] to [0132] & US 2011/0271084 A1	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
19 November, 2013 (19.11.13)

Date of mailing of the international search report
26 November, 2013 (26.11.13)

Name and mailing address of the ISA/
Japanese Patent Office

Authorized officer

Facsimile No.

Telephone No.

A. 発明の属する分野の分類（国際特許分類（I P C））

Int.Cl. G06F12/00(2006.01)i, G06F17/30(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（I P C））

Int.Cl. G06F12/00, G06F17/30

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	1 9 2 2 - 1 9 9 6 年
日本国公開実用新案公報	1 9 7 1 - 2 0 1 3 年
日本国実用新案登録公報	1 9 9 6 - 2 0 1 3 年
日本国登録実用新案公報	1 9 9 4 - 2 0 1 3 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	WO 2012/108175 A1（日本電気株式会社）2012.08.16, [0013]-[0030] （ファミリーなし）	1-10
A	JP 2011-233033 A（富士通株式会社）2011.11.17, 段落 0013-15, 0028, 0055-0132 & US 2011/0271084 A1	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

1 9 . 1 1 . 2 0 1 3

国際調査報告の発送日

2 6 . 1 1 . 2 0 1 3

国際調査機関の名称及びあて先

日本国特許庁（I S A / J P）
郵便番号 1 0 0 - 8 9 1 5
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）

田川 泰宏

5 U

4 2 3 6

電話番号 0 3 - 3 5 8 1 - 1 1 0 1 内線 3 5 6 5