(19)日本国特許庁(JP)　(12)特　許　公　報(B2)　(11)特許番号
特許第6268116号
(P6268116)
(45)発行日　平成30年1月24日(2018.1.24)　(24)登録日　平成30年1月5日(2018.1.5)

(51)Int.Cl.　F I
G06F 12/00 (2006.01)　G06F 12/00 560G

請求項の数 16　(全 21 頁)

(21)出願番号　特願2015-58741(P2015-58741)
(22)出願日　平成27年3月20日(2015.3.20)
(65)公開番号　特開2016-177688(P2016-177688A)
(43)公開日　平成28年10月6日(2016.10.6)
審査請求日　平成29年3月7日(2017.3.7)

(73)特許権者　317006041
東芝メモリ株式会社
東京都港区芝浦一丁目１番１号
(74)代理人　100091982
弁理士　永井　浩之
(74)代理人　100091487
弁理士　中村　行孝
(74)代理人　100082991
弁理士　佐藤　泰和
(74)代理人　100105153
弁理士　朝倉　悟
(74)代理人　100107582
弁理士　関根　毅
(74)代理人　100118843
弁理士　赤岡　明

最終頁に続く

(54)【発明の名称】データ処理装置、データ処理方法およびコンピュータプログラム

(57)【特許請求の範囲】
【請求項１】
キーを指定した読み出し要求を受け付け、前記読み出し要求を解釈する要求解釈部と、
複数のキーを所定のルールに従った順序で保持する順序キー列における前記キーに対して前記読み出し要求で指示された位置関係にあるキーを特定する第１アクセス部と、
第１アドレスと第２アドレスとを対応づけて管理する内部または外部の第１記憶装置から、前記第１アクセス部で特定されたキーのハッシュ値に基づく第１アドレスに対応する第２アドレスを取得する第２アクセス部と、
内部または外部の第２記憶装置において前記第２アクセス部で取得された前記第２アドレスに格納されたデータを読み出す第３アクセス部と、
を備えたデータ処理装置。
【請求項２】
前記要求解釈部で解釈された前記読み出し要求の内容に応じて、前記第１アクセス部に前記キーと前記読み出し要求で指示された位置関係の情報とを通知し、前記第１アクセス部で特定されたキーを受け取り、受け取ったキーを前記第２アクセス部に通知する処理振分部
をさらに備えた請求項１に記載のデータ処理装置。
【請求項３】
前記第２記憶装置の第２アドレスを管理するアドレス割当部を備え、
前記要求解釈部は、キーとバリューとを指定した書き込み要求を受け付け、前記書き込

み要求を解釈し、
前記第１アクセス部は、前記書き込み要求で指定されたキーを前記順序キー列に追加することにより前記順序キー列を更新し、
前記アドレス割当部は、前記書き込み要求で指定されたキーに対して第２アドレスを割り当て、
前記第２アクセス部は、前記第１記憶装置において前記指定されたキーのハッシュ値に基づく第１アドレスに対応づけて前記第２アドレスを登録し、
前記第３アクセス部は、前記第２記憶装置における前記第２アドレスに前記バリューを含むデータを書き込む
請求項１または２に記載のデータ処理装置。
【請求項４】
前記順序キー列は、前記第２記憶装置に格納されており、
前記第１記憶装置は、内部キーのハッシュ値に基づく第１アドレスと、前記順序キー列が格納された第２アドレスとの対応を管理し、
前記第２アクセス部は、前記第１記憶装置から前記内部キーに対応する前記第２アドレスを取得し、
前記第３アクセス部は、前記第２アクセス部で取得された前記第２アドレスに格納されたデータである前記順序キー列を前記第２記憶装置から読み出し、
前記第１アクセス部は、前記第３アクセス部により読み出された前記順序キー列を用いて前記キーを特定する
請求項１ないし３のいずれか一項に記載のデータ処理装置。
【請求項５】
前記順序キー列は、前記第２記憶装置に格納されており、
前記第１記憶装置は、内部キーのハッシュ値に基づく第１アドレスと、前記順序キー列が格納された第２アドレスとの対応を管理し、
前記第２アクセス部は、前記第１記憶装置から前記内部キーに対応する前記第２アドレスを取得し、
前記第３アクセス部は、前記第２アクセス部で取得された前記第２アドレスに格納されたデータに含まれる前記順序キー列を前記第２記憶装置から読み出し、
前記第１アクセス部は、前記第３アクセス部により読み出された前記順序キー列に、前記書き込み要求で指定されたキーを追加することにより前記順序キー列を更新し、
前記アドレス割当部は、前記更新された順序キー列に対して第２アドレスを割り当て、
前記第２アクセス部は、前記第１記憶装置において前記内部キーのハッシュ値に基づく第１アドレスに対応づけて前記割り当てられた第２アドレスを登録し、
前記第３アクセス部は、前記第２記憶装置における前記割り当てられた第２アドレスに前記順序キー列を含むデータを書き込む
請求項３に記載のデータ処理装置。
【請求項６】
前記内部キーを生成する内部キー生成部
をさらに備えた請求項４または５に記載のデータ処理装置。
【請求項７】
前記順序キー列を複数に分割した各部分に、それぞれ異なる内部キーが割り当てられ、
前記内部キー生成部は、前記読み出し要求または前記書き込み要求で指定されたキーが、前記複数の各部分のいずれに属するかに応じて前記内部キーを生成する
請求項６に記載のデータ処理装置。
【請求項８】
前記順序キー列は、前記複数のキーにそれぞれ対応づけて前記第２アドレスを管理し、
前記第１アクセス部は、前記特定したキーに対応する前記第２アドレスを特定し、
前記第３アクセス部は、前記第２記憶装置における前記特定した第２アドレスからデータを読み出す

請求項１ないし７のいずれか一項に記載のデータ処理装置。
【請求項９】
前記順序キー列は、前記複数のキーを辞書順で管理する
請求項１ないし８のいずれか一項に記載のデータ処理装置。
【請求項１０】
前記読み出し要求は、１つのキーを指定し、前記キーの次の位置のキー、前記キーの前の位置のキー、またはこれらの両方を読み出すことを要求する、
もしくは、
前記読み出し要求は、２つのキーを含み、前記２つのキーの間に位置するキーの読み出しを要求する
請求項１ないし９のいずれか一項に記載のデータ処理装置。
【請求項１１】
キーを指定した読み出し要求を受け付け、前記読み出し要求を解釈する要求解釈ステップと、
複数のキーを所定のルールに従った順序で保持する順序キー列における前記キーに対して前記読み出し要求で指示された位置関係にあるキーを特定する第１アクセスステップと、
第１アドレスと第２アドレスとを対応づけて管理する内部または外部の第１記憶装置から、前記第１アクセス部で特定されたキーのハッシュ値に基づく第１アドレスに対応する第２アドレスを取得する第２アクセスステップと、
内部または外部の第２記憶装置において前記第２アクセス部で取得された前記第２アドレスに格納されたデータを読み出す第３アクセスステップと、
をコンピュータが実行するデータ処理方法。
【請求項１２】
前記要求解釈ステップで解釈された前記読み出し要求の内容に応じて、前記第１アクセスステップに前記キーと前記読み出し要求で指示された位置関係の情報とを通知し、前記第１アクセスステップで特定されたキーを受け取り、受け取ったキーを前記第２アクセスステップに通知する処理振分ステップ
をさらに備えた請求項１１に記載のデータ処理方法。
【請求項１３】
前記第２記憶装置の第２アドレスを管理するアドレス割当ステップを備え、
前記要求解釈ステップは、キーとバリューとを指定した書き込み要求を受け付け、前記書き込み要求を解釈し、
前記第１アクセスステップは、前記書き込み要求で指定されたキーを前記順序キー列に追加することにより前記順序キー列を更新し、
前記アドレス割当ステップは、前記書き込み要求で指定されたキーに対して第２アドレスを割り当て、
前記第２アクセスステップは、前記第１記憶装置において前記指定されたキーのハッシュ値に基づく第１アドレスに対応づけて前記第２アドレスを登録し、
前記第３アクセスステップは、前記第２記憶装置における前記第２アドレスに前記バリューを含むデータを書き込む
請求項１１または１２に記載のデータ処理方法。
【請求項１４】
キーを指定した読み出し要求を受け付け、前記読み出し要求を解釈する要求解釈ステップと、
複数のキーを所定のルールに従った順序で保持する順序キー列における前記キーに対して前記読み出し要求で指示された位置関係にあるキーを特定する第１アクセスステップと、
第１アドレスと第２アドレスとを対応づけて管理する内部または外部の第１記憶装置から、前記第１アクセス部で特定されたキーのハッシュ値に基づく第１アドレスに対応する

第２アドレスを取得する第２アクセスステップと、
内部または外部の第２記憶装置において前記第２アクセス部で取得された前記第２アドレスに格納されたデータを読み出す第３アクセスステップと、
をコンピュータに実行させるためのコンピュータプログラム。
【請求項１５】
前記要求解釈ステップで解釈された前記読み出し要求の内容に応じて、前記第１アクセスステップに前記キーと前記読み出し要求で指示された位置関係の情報とを通知し、前記第１アクセスステップで特定されたキーを受け取り、受け取ったキーを前記第２アクセスステップに通知する処理振分ステップ
をさらに備えた請求項１４に記載のコンピュータプログラム。
【請求項１６】
前記第２記憶装置の第２アドレスを管理するアドレス割当ステップをさらに前記コンピュータに実行させ、
前記要求解釈ステップは、キーとバリューとを指定した書き込み要求を受け付け、前記書き込み要求を解釈し、
前記第１アクセスステップは、前記書き込み要求で指定されたキーを前記順序キー列に追加することにより前記順序キー列を更新し、
前記アドレス割当ステップは、前記書き込み要求で指定されたキーに対して第２アドレスを割り当て、
前記第２アクセスステップは、前記第１記憶装置において前記指定されたキーのハッシュ値に基づく第１アドレスに対応づけて前記第２アドレスを登録し、
前記第３アクセスステップは、前記第２記憶装置における前記第２アドレスに前記バリューを含むデータを書き込む
請求項１４または１５に記載のコンピュータプログラム。
【発明の詳細な説明】
【技術分野】
【０００１】
本発明の実施形態は、データ処理装置、データ処理方法およびコンピュータプログラムに関する。
【背景技術】
【０００２】
近年、Ｗｅｂメールやソーシャルネットワークなどに代表される高度なＷｅｂサービスが急速に普及している。このようなＷｅｂサービスでは、ＷｅｂサーバがユーザからのＷｅｂページ表示要求を受けると、Ｗｅｂサーバは必要な個々のテキスト、画像や動画などのデータを、逐一バックエンドのストレージに問い合わせて取得し、ユーザに応答する。このようなストレージとしては、従来からのファイルシステムベースのデータベースも使われているが、近年では、一意のＵＲＬを指定してコンテンツデータにアクセスするという、インターネット上のコンテンツアクセス方式と親和性の高いオブジェクトストレージが、活用され始めている。このようなオブジェクトストレージでは、Ｗｅｂサーバといったクライアント装置から、イーサネット（登録商標）越しに可変長の一意のＫｅｙ（キー）で、可変長のＶａｌｕｅ（バリュー）にアクセスするというキーバリューストア（ＫＶＳ）型のＩ／Ｆがとられている。Ｗｅｂサーバの場合、キーはＵＲＬ、バリューはコンテンツデータであることが一般的である。キーに関して、オブジェクトＩＤが可変長でなく、固定長になるものもあるが、基本的な原理等は同様である。ストレージデバイスとしてはＨＤＤやＳＳＤが用いられる。
【０００３】
最もシンプルなアクセスの方法は、例えばＫｅｙ＝“００１”に対する読み出しアクセスは、ＧＥＴ（Ｋｅｙ＝“００１”）など、読み出したいバリューに対応するキーを指定した要求を送り、応答としてバリューを受けるものである。また、書き込みアクセスは、ＰＵＴ（Ｋｅｙ＝“００２”，Ｖａｌｕｅ＝“ＧＨＩ”）など、書き込みを行いたいキー

とバリューを指定して、要求を送るという形態である。なお、応答として、指定したキーに対応するバリューのみを受ける形態、指定したキーと、対応するバリューとの両方を受ける形態など種々の態様がある。

【０００４】

より高度なアクセスとして、指定したキーの辞書順で、次の／前のキーに対応するバリューを読み出す、ＧＥＴＮＥＸＴ要求／ＧＥＴＰＲＥＶ要求などもある。例えば（Ｋｅｙ，Ｖａｌｕｅ）＝（“０００”，“ＡＢＣ”）、（“００２”，“ＤＥＦ”）、（“００４”，“ＧＨＩ”）、（“００５”，“ＪＫＬ”）という、４つのキーおよびバリューが格納されているとする。このとき、ＧＥＴＮＥＸＴ（Ｋｅｙ＝“００２”）の要求を送ると、応答として次のキーであるＫｅｙ＝“００４”に対応するバリューである“ＧＨＩ”が返される。ＧＥＴＰＲＥＶ（Ｋｅｙ＝“００２”）の場合は、前のキーであるＫｅｙ＝“０００”に対応するバリューである“ＡＢＣ”が返される。

【０００５】

このような要求は、大きなデータを複数のバリューに分割して、各バリューにキーを対応付けておき、各キーに対応するバリューを順次読み出す場合などに用いられる。なお、要求で指定されたキーが存在しない場合もある。例えばＫｅｙ＝“００１”などのような場合もある。その場合でも、ＧＥＴＰＲＥＶ（Ｋｅｙ＝“００１”）に対する応答は、Ｋｅｙ＝“０００”に対応するバリューである“ＡＢＣ”であり、ＧＥＴＮＥＸＴ（Ｋｅｙ＝“００１”）に対する応答は、Ｋｅｙ＝“００２”に対応するバリューである“ＤＥＦ”となる。

【０００６】

同様に２つのキーの間に存在する全てのキーおよびバリューを応答するＧＥＴＲＡＮＧＥ要求もある。例えばＧＥＴＲＡＮＧＥ（Ｋｅｙ１＝“００１”，Ｋｅｙ２＝“００４”）の要求を送ると、（Ｋｅｙ，Ｖａｌｕｅ）＝（“００２”，“ＤＥＦ”）、（“００４”，“ＧＨＩ”）が応答される。この例では、応答としてバリューと、キーとの両方が返されている。

【０００７】

キーバリューストア（ＫＶＳ）の典型的な形態として、ハッシュテーブルベースのものが挙げられる。ハッシュテーブルベースのＫＶＳは、キーに対応するバリューのストレージ上のアドレス（以下、ＫＶアドレス）を、ハッシュテーブルで管理する。典型的には、キーを入力とする所定のハッシュ関数を計算し、ハッシュテーブル上のそのハッシュ値（通常、キーよりも十分短いバイト列である）に基づくアドレス（以下、ハッシュアドレス）のエントリに、バリューが記憶されているストレージ上のＫＶアドレスを格納しておく。なお、ハッシュ値から、ハッシュテーブル上のハッシュアドレスが一意に決まる限り、ハッシュ値とハッシュアドレスとが必ずしも一致する必要はないが、以下では説明のためハッシュ値とハッシュアドレスが一致する場合を想定する。このようにすることによって、ＧＥＴ要求（バリューの読み出し）などを受けた場合、要求で指定されたキーに対して、ハッシュ関数を用いて瞬時にバリューが格納されているストレージ上のＫＶアドレスを特定することができる。よって、高速にバリューにアクセスして、応答を返すことができる。

【０００８】

しかしながら、ハッシュテーブルベースのＫＶＳは、上記のＧＥＴＮＥＸＴ要求／ＧＥＴＰＲＥＶ要求／ＧＥＴＲＡＮＧＥ要求に対する応答を返すことが出来ない。なぜなら、ハッシュテーブルベースのＫＶＳは、指定されたキーのバリューを読み出すことはできるものの、指定されたキーの前後にどのようなキーが格納されているかを知る事が出来ないためである。各キーはハッシュ関数を介してハッシュ値に変換されて、ハッシュテーブル上のハッシュアドレスとして扱われる。ハッシュテーブル上のハッシュアドレスの前後関係は、実際のキーの前後関係とは異なるものであり、ハッシュテーブルからは実際のキーの前後関係を辿ることができない。よって、所定のキーの辞書順で、指定されたキーの次のキーを得るには、全検索を行うしかなく、それは現実的ではない。

【０００９】
上記のようなＧＥＴＮＥＸＴ要求／ＧＥＴＰＲＥＶ要求／ＧＥＴＲＡＮＧＥ要求などに応答するため、キーを辞書順でアクセス可能なＫＶＳ（以下、順序型ＫＶＳ）がある。順序型ＫＶＳは、ハッシュテーブルではなく、辞書順でアクセス可能なデータ構造でキーを管理し、ＧＥＴＮＥＸＴ要求／ＧＥＴＰＲＥＶ要求／ＧＥＴＲＡＮＧＥ要求などに対応するものである。オープンソース・ソフトウェアとしては、ＬｅｖｅｌＤＢ、ＲｏｃｋｓＤＢなどが有名である。このような順序型ＫＶＳは、Ｂ－ｔｒｅｅやＬＳＭ－ｔｒｅｅなどのデータ構造でキーを管理し、通常のＧＥＴ要求などに対して、上記データ構造上でキーの探索を行う。探索の結果、キーが見つかった場合は、キーに対応するストレージ上のバリューのＫＶアドレスを取得し、取得したＫＶアドレスに基づきストレージ上のバリューにアクセスして、そのバリューを応答する。上記データ構造では、キーを辞書順で並べて管理しているため、ＧＥＴＮＥＸＴ要求を受けた場合も、通常のＧＥＴ要求と同様に、まず指定されたキーを探索し、その後にその次のキーを辿って、当該次のキーに対応するバリューのＫＶアドレスにアクセスすることができる。よって、ＧＥＴＮＥＸＴ要求にも応答可能である。

【００１０】
前述の順序型ＫＶＳは、ＧＥＴＮＥＸＴ要求等に応答できる機能を持つものの、ハッシュテーブルベースのものと異なり、キーを「探索」する必要があるという問題がある。応答速度を高めるために数々のデータベース技術によって改良が試みられているものの、順序型ＫＶＳでは、基本的に探索が不要なハッシュテーブルベースのＫＶＳに比べて、キーを特定するまでの手順が複雑になるという問題がある。このため目的のバリューを検索するまでの時間が長くなり、結果的に応答速度の悪化につながる。

【先行技術文献】
【特許文献】
【００１１】
【特許文献１】米国出願公開第２０１１／０２７６７４４号明細書

【発明の概要】
【発明が解決しようとする課題】
【００１２】
本発明の実施形態は、指定されたキーとある位置関係にあるキーに対するデータを高速に検索可能することを目的とする。

【課題を解決するための手段】
【００１３】
本発明の実施形態としてのデータ処理装置は、要求解釈部と、第１アクセス部と、第２アクセス部と、第３アクセス部とを、備える。前記要求解釈部は、キーを指定した読み出し要求を受け付け、前記読み出し要求を解釈する。前記第１アクセス部は、複数のキーを所定のルールに従った順序で保持する順序キー列における前記キーに対して前記読み出し要求で指示された位置関係にあるキーを特定する。前記第２アクセス部は、第１アドレスと第２アドレスとを対応づけて管理する内部または外部の第１記憶装置から、前記第１アクセス部で特定されたキーのハッシュ値に基づく第１アドレスに対応する第２アドレスを取得する。前記第３アクセス部は、内部または外部の第２記憶装置において前記第２アクセス部で取得された前記第２アドレスに格納されたデータを読み出す。

【図面の簡単な説明】
【００１４】
【図１】本発明の実施形態に係るデータ処理システムを示す図。
【図２】第１の実施形態に係るデータ処理装置のブロック図。
【図３】ハッシュテーブル、順序キー列、およびバリューを含むデータ群を示す図。
【図４】ハッシュテーブルの他の例を示す図。
【図５】第１の実施形態の動作例のフローチャート。
【図６】第１の実施形態の他の動作例のフローチャート。

【図７】第２の実施形態に係るデータ処理装置のブロック図。
【図８】順序キー列をバリューとして保存する様子を示す図。
【図９】第２の実施形態の動作例のフローチャート。
【図１０】第２の実施形態の他の動作例のフローチャート。
【発明を実施するための形態】
【００１５】
以下、図面を参照しながら、本発明の実施形態について説明する。
【００１６】
図１は、本発明の実施形態に係るデータ処理システムを示す。データ処理システムは、データ処理装置１１と、複数のクライアント装置２１～２３を備えている。
【００１７】
データ処理装置１１は、イーサネットなどのネットワーク越しに、クライアント装置２１、２２、２３からの各種読み出しまたは書き込みの要求を受け付けるキーバリューストア装置である。クライアント装置２１～２３は、例えばＷｅｂサーバ等のサーバであり、Ｗｅｂサーバの場合、一例として、ＵＲＬをキー、Ｗｅｂページ等のコンテンツデータをバリューとして、キーとバリュー（キーバリュー）の書き込み要求をデータ処理装置１１に行う。クライアント装置２１～２３は、インターネット等のネットワークを介してユーザ端末と接続されており、例えば、ユーザ端末からＵＲＬを指定した、コンテンツデータの書き込み指示を受ける。クライアント装置２１～２３は、この書き込み指示に基づき、ＵＲＬをキー、コンテンツデータをバリューとして、キーバリューの書き込み要求を生成して、データ処理装置１１に送る。データ処理装置１１は、キーバリューの書き込み要求を受けると、後述する各実施形態に従ってキーバリューの書き込み処理を行う。書き込みが完了すると、書き込み完了の応答をクライアント装置２１～２３に返す。
【００１８】
また、クライアント装置２１～２３は、ユーザ端末から、一例として、キーを指定した、バリューの読み出し指示を受けると、キーを指定した、キーバリューの読み出し要求を生成してデータ処理装置１１に送る。データ処理装置１１は、クライアント装置２１～２３からの読み出し要求を受けて、指定されたキーに基づき、後述する各実施形態に従って読み出し処理を行ってバリューを取得し、取得したバリューを含む応答をクライアント装置２１～２３に返す。以下、このようなデータ処理装置の各種の実施形態について説明する。
【００１９】
（第１の実施形態）
図２は、本発明の第１の実施形態に係るデータ処理装置を示す。
【００２０】
図２のデータ処理装置は、要求解釈部１０１、処理振分部１０２、ハッシュテーブルアクセス部１０３、順序キー列アクセス部１０４、ＫＶアドレス割当部１０５、ＫＶアクセス部１０６、応答生成部１０７、ハッシュテーブル記憶部１０８、順序キー列記憶部１０９、およびＫＶ記憶部１１０を備える。順序キー列アクセス部１０４は一例として第１アクセス部に対応し、ハッシュテーブルアクセス部１０３は一例として第２アクセス部に対応し、ＫＶアクセス部１０６は一例として第３アクセス部に対応する。このデータ処理装置は、典型的な形態としては、ＣＰＵ、メモリ、ストレージ、ネットワークインタフェースなどで構成される。一部の機能がＦＰＧＡやＡＳＩＣなどの専用ハードウェアによって実現されていてもよい。
【００２１】
要求解釈部１０１は、ＧＥＴ要求、ＧＥＴＮＥＸＴ要求、ＧＥＴＰＲＥＶ要求、または、ＧＥＴＲＡＮＧＥ要求またはＰＵＴ要求などの要求を受け付け、受信した要求を解釈する。ＧＥＴ要求は、指定したキーに対応するバリューを読み出すことの要求である。ＧＥＴＮＥＸＴ要求は、指定したキーの辞書順で、次のキーに対応するバリューを読み出すことの要求である。ＧＥＴＰＲＥＶ要求は、指定したキーの辞書順で、前のキーに対応する

バリューを読み出すことの要求である。ＧＥＴＲＡＮＧＥ要求は，２つのキーを指定し、当該指定した２つのキーの間に辞書順で存在する全てのキーに対応するバリューを読み出すことの要求である。ＰＵＴ要求は、キーとバリューとを指定し、キーに対応づけてバリューを書き込むことの要求である。

【００２２】

ＫＶ記憶部１１０は、キーとバリューとを含むデータを記憶している。キー毎に対応して、キーバリューが格納されている。図３の右下において複数のキーに対応する複数のデータが格納されている様子が示される。例えば、ＫＶ（Ｂｒｕｃｅ）は、キーであるＢｒｕｃｅと、当該キーに対応するバリューとを含むデータを表している。図３では、複数のデータ（キーバリュー）が、連続したアドレスに書き込まれている様子が示されるが、これは一例であり、どのように格納するかは、書き込むＫＶアドレスを割り当てる方式等に依存する。本例では、データは、キーとバリュー（すなわちキーバリュー）を含むが、データが、バリューを含み、キーは含まない構成もあり得る。またデータが、キーとバリュー以外の情報（例えばバリューのサイズ情報など）を含んでもかまわない。以下では、データが、少なくともバリューを含むことを前提として、説明を続ける。ＫＶ記憶部１１０は、ハードディスクまたはＳＳＤ、あるいは任意の不揮発性メモリ、あるいはその他の任意の永続的にデータを記憶する装置で構成されることができる。

【００２３】

ハッシュテーブル記憶部１０８は、ハッシュテーブルを保持する。ハッシュテーブルとは、キーを所定のハッシュ関数にかけて算出されたハッシュ値に基づくアドレス（以下、ハッシュアドレス）のエントリに、そのキーに関わる情報、特にバリューを含むデータのＫＶ記憶部１１０上のＫＶアドレス（以下、ＫＶアドレス）等を格納するテーブルである。ハッシュアドレスは、ハッシュ値そのものでもよいし、ハッシュ値から一意に特定される別の値でもかまわない。本例では、ハッシュアドレスが、ハッシュ値そのものである場合を想定する。図３の左にハッシュテーブルの例を示す。Ｈ１、Ｈ５等はハッシュ値を表している。ＡＤＲ（Ｂｒｕｃｅ）は、キーであるＢｒｕｃｅに対応するデータが格納された記憶部上のＫＶアドレスを表す。ハッシュテーブル記憶部１０８は、ＳＳＤまたはＤＲＡＭなど、任意の記憶媒体により構成でき、メモリで構成する場合、不揮発性メモリでも揮発性メモリでもよい。

【００２４】

順序キー列記憶部１０９は、順序キー列を管理する。順序キー列は、複数のキーを所定のルールに従った順序で保持したものである。例えば、順序キー列は、Ｂ－ｔｒｅｅまたはＬＳＭ－ｔｒｅｅなどのデータ構造であり、複数のキーを辞書順の並びで保持する。図３の右上に順序キー列の例を示す。図３の例では、Ｂｏｂ、Ｂｒｕｃｅ、Ｊｏｎｅ、Ｋｅｎがこの順序で並んでいる。このデータ構造は、ハッシュテーブルと異なり、指定されたキーの前後のキーに容易にアクセスできるという特長を持つ。順序キー列は、単純にキーの並びだけ表してもよいし、キーのみならず、そのキーに関わる情報、特にバリューのＫＶアドレスをキーに関連付けて含んでも良い。以降では簡単のため、ＫＶアドレスをキーに関連付けて含まない例を用いてＫＶアドレスを得るためにハッシュテーブルにアクセスする実施の形態を説明する。しかしながら、順序キー列にアクセスした後にハッシュテーブルにアクセスしてＫＶアドレスを得る代わりに、順序キー列内にＫＶアドレスをキーに関連付けて含むことで、その後のハッシュテーブルアクセスを省略する形態でもよい。順序キー列の構成は、具体的には様々な形態があり得るが、そのアルゴリズム自体は本実施形態とは無関係であり、本実施形態としては何でも良い。なお、順序キー列の管理には、一例としてファイルシステムが利用され、個々の部分データは個々のファイルとして実装される。順序キー列記憶部１０９は、ＨＤＤ、ＳＳＤまたはＤＲＡＭなど、任意の記憶媒体により構成でき、メモリで構成する場合、不揮発性メモリでも揮発性メモリでもよい。

【００２５】

なお、図３では、ＫＶ記憶部１１０に格納されているデータに対し矢印付きの線が引かれている。当該線は、ハッシュテーブルのエントリに格納されているＫＶアドレスと、順

序キー列のキー（エントリ）とからそれぞれ引かれている。これは、ハッシュテーブルのエントリに格納されたＫＶアドレスが指す領域に、当該エントリに対応するキーに対するデータが格納されていることを表現している。

【００２６】

ＫＶ記憶部１１０、ハッシュテーブル記憶部１０８および順序キー列記憶部１０９はそれぞれ別々のハードウェアでもよいし、共通のハードウェアでもよい。またはこれらの内の一部が同じハードウェアで、残りが別のハードウェアでもよい。

【００２７】

処理振分部１０２は、要求解釈部１０１で解釈された要求の内容に応じて、ハッシュテーブルアクセス部１０３または順序キー列アクセス部１０４またはこれらの両方に、処理を依頼する。

【００２８】

ハッシュテーブルアクセス部１０３は、指定されたキーのハッシュ値を計算し、当該ハッシュ値に対応するハッシュアドレスに基づき、ハッシュテーブルのエントリへアクセスし、エントリに格納されているＫＶ記憶部１１０上のＫＶアドレスを取得する。

【００２９】

ＫＶアドレス割当部１０５は、ＫＶ記憶部１１０に新規にデータ（バリューまたはキーバリューなど）を書き込むときのＫＶアドレスを割り当てる。ＫＶアドレスの割り当て方法は、ログ形式でＰＵＴ要求が来た順番にシーケンシャルにＫＶアドレスを割り当てて、その順番で書き込みを行う方式が一般的であるが、これに限定されるものではない。

【００３０】

ＫＶアクセス部１０６は、ＫＶ記憶部１１０にアクセスして、ハッシュテーブルアクセス部１０３（または順序キー列アクセス部１０４）によって取得されたＫＶアドレスに格納されているデータを取得する。

【００３１】

応答生成部１０７は、ＫＶアクセス部１０６により取得されたデータ（バリューまたはキーバリューなど）に基づき応答を生成して、生成した応答を、要求解釈部１０１で受けた要求の送信元に送信する。

【００３２】

ここでハッシュテーブルについて補足説明を行う。前述したハッシュテーブルの構成方法には、様々な形態があり得る。特に異なるキーが同じハッシュ値をとってしまうという、いわゆる衝突が発生した場合に、どのようにその同じハッシュ値を持つ複数のキーのエントリを格納するかについては、様々なアルゴリズムがある。

【００３３】

例えば、衝突した場合には、所定のルールに基づいて他の空いているエントリが見つかるまで探索する開番地法、一つのエントリに、例えばリスト構造などで複数のエントリを格納する連鎖法などがある。別の例として、指定されたキーと、対応するエントリが一致するかどうかを確認する方法がある。この方法の場合、ハッシュテーブルの各エントリに、ＫＶアドレスに加えて、キーそのものを格納して、指定されたキーと、エントリに格納されたキーの一致性を確認してもよい。

【００３４】

ハッシュテーブルの容量を減らすために、ハッシュテーブルの各エントリに、キーを所定の関数で縮退させたビット列をＫＶアドレスに加えて格納してもよい。例えばキーであるＢｒｕｃｅを関数Ｓｉｇ（）で縮退させてビット列Ｓｉｇ（Ｂｒｕｃｅ）を、ＡＤＲ（Ｂｒｕｃｅ）に加えてエントリに格納する。縮退させたビット列は、縮退キーとも呼ぶ。このハッシュテーブルの例を図４に示す。この場合は、最終的に、指定されたキーとの完全マッチングを確認するために、ＫＶ記憶部１１０上にキーバリューを格納し、そのキーバリューにアクセスして、そこに含まれるキーとの最終的なマッチングを図る必要がある。すなわち、まずは、指定されたキーを縮退させてビット列を取得し、そのビット列と、ハッシュテーブルの該当するエントリに格納されている、縮退させたビット列とのマッチ

ングを確認する。この段階でマッチングしていなければ、マッチしていない（つまり当該エントリに含まれるＫＶアドレスに格納されているバリューは、指定されたキーに対応するバリューではない）と判断し、エントリの探索を再開する。マッチしていれば、マッチしている可能性は高いが、完全にはマッチすることはまだ確認されていないため、そのエントリに格納されているＫＶ記憶部１１０上のＫＶアドレスを元にＫＶ記憶部１１０にアクセスしてキーバリューを読み出す。そして、そこに含まれているキーと、指定されたキーとのマッチングを確認する。マッチしたら、そのときに読み出したバリューをそのまま採用し、マッチしなかった場合は再びハッシュテーブルに戻り、エントリの探索を再開し、同様にマッチするキーが見つかるか、あるいはマッチするキーが無いことが確認できるまで繰り返す。

【００３５】

このようなハッシュテーブルの構造、および探索アルゴリズム自体は、本実施形態の本質とは無関係であり、本実施形態では何でもよい。以下の説明では、ＫＶ記憶部１１０上に少なくともバリューを含むデータを格納することを仮定するが、本実施形態は、これに限定するものではない。

【００３６】

本実施形態は、要求解釈部１０１が受けた要求の種類によって、ハッシュテーブルと順序キー列とを使い分けるところを特徴の１つとする。以下、ＧＥＴ要求を受けたとき、ＧＥＴＮＥＸＴ要求／ＧＥＴＰＲＥＶ要求／ＧＥＴＲＡＮＧＥ要求を受けたとき、および、ＰＵＴ要求を受けたときの動作について、それぞれ説明する。

【００３７】

まず、キーを指定したＧＥＴ要求を受けたときの動作を説明する。要求解釈部１０１が、その要求がＧＥＴであることを解釈し、処理振分部１０２は、キーをハッシュテーブルアクセス部１０３に渡す。ハッシュテーブルアクセス部１０３は、所定のハッシュ関数を用いて、指定されたキーのテーブルエントリにアクセスし、ＫＶアドレスを取得する。例えば指定されたキーのハッシュ値を計算し、ハッシュ値に対応するハッシュテーブル上のエントリを特定する。そして、特定したエントリに格納されているＫＶアドレスに基づきＫＶアクセス部１０６が、ＫＶ記憶部１１０にアクセスして、ＫＶ記憶部１１０に記憶されているデータ（キーバリュー）を読み出す。読み出したキーバリューに含まれるキーが、指定されたキーにマッチすることを確認した上で、応答生成部１０７が、読み出したバリューを含む応答を生成して、ＧＥＴ要求の送信元に送信する。読み出したキーバリューに含まれるキーが、指定されたキーにマッチするかの判断は、ＫＶアクセス部１０６が行っても良いし、処理振分部１０２が行っても良いし、当該判断を行う処理部を独立して設けてもよい。指定されたキーにマッチしない場合は、マッチしなかった場合は再びハッシュテーブルに戻り、エントリの探索を再開し、同様にマッチするキーが見つかるか、あるいはマッチするキーが無いことが確認できるまで繰り返す。

【００３８】

次に、ＧＥＴＮＥＸＴ要求、ＧＥＴＰＲＥＶ要求、または、ＧＥＴＲＡＮＧＥ要求を受けた場合の動作を説明する。ＧＥＴＮＥＸＴ要求は、指定したキーの辞書順で、次のキーに対応するバリューを読み出すことの要求である。ＧＥＴＰＲＥＶ要求は、指定したキーの辞書順で、前のキーに対応するバリューを読み出すことの要求である。ＧＥＴＲＡＮＧＥ要求は，２つのキーを指定し、当該指定した２つのキーの間に存在する全てのキー（当該範囲を規定する２つのキーを含むように定義しても、含まないように定義してもよい）に対応するバリューを読み出すことの要求である。

【００３９】

処理振分部１０２が、ＧＥＴＮＥＸＴ要求、ＧＥＴＰＲＥＶ要求、または、ＧＥＴＲＡＮＧＥ要求を受けたことを解釈し、順序キー列アクセス部１０４に、指定されたキーと、要求の内容を通知する。

【００４０】

順序キー列アクセス部１０４は、ＧＥＴＮＥＸＴ要求であれば、指定されたキーの次の

キー、ＧＥＴＰＲＥＶ要求であれば、指定されたキーの前のキー、ＧＥＴＲＡＮＧＥ要求であれば、指定された２つのキーの間にある全てのキーを、順序キー列記憶部１０９内の順序キー列に基づき、特定する。なお、前述したように、順序キー列において、キーと関連づけて、バリューが格納されているＫＶ記憶部１１０上のＫＶアドレスを保持してもよく、この場合は、特定したキーに対応するバリューのＫＶ記憶部１１０上のＫＶアドレスも特定してもよい。

【００４１】

順序キー列アクセス部１０４は、特定したキーを、処理振分部１０２に渡し、処理振分部１０２は、当該特定したキーをハッシュテーブルアクセス部１０３に渡す。ハッシュテーブルアクセス部１０３は、ハッシュテーブルを用いて、当該特定したキーに対応するバリューのＫＶ記憶部１１０上のＫＶアドレスを特定する。なお、順序キー列に関連づけられたＫＶアドレスを特定した場合は、順序キー列アクセス部１０４は、特定したＫＶアドレスをＫＶアクセス部１０６に渡せばよい。

【００４２】

この後の処理は、ＧＥＴ要求の場合と同様である。すなわち、ＫＶアクセス部１０６が、ＫＶ記憶部１１０にアクセスしてバリュー（もしくはキーバリューなど）を読み出し、応答生成部１０７が、読み出したバリュー（もしくはキーバリューなど）を含めた応答を生成して、要求の送信元に送信する。

【００４３】

次に、ＰＵＴ要求を受けた場合の動作を説明する。ＰＵＴ要求は、キーとバリューとを指定し、キーに対応づけてバリューを書き込むことの要求である。

【００４４】

処理振分部１０２が、ＰＵＴ要求を受けたことを解釈し、ハッシュテーブルアクセス部１０３にキーとバリューとを渡し、順序キー列アクセス部１０４にキーを渡す。ハッシュテーブルアクセス部１０３は、指定されたキーのハッシュ値を計算し、ハッシュテーブル上の当該ハッシュアドレスに対応するエントリを生成する。ＫＶアドレス割当部１０５は、指定されたキーとバリュー（少なくともバリュー）とを含むデータを格納するＫＶ記憶部１１０上のＫＶアドレスを、当該キーに対して割り当てる。ハッシュテーブルアクセス部１０３は、生成したエントリに、ＫＶアドレス割当部１０５が決定したＫＶ記憶部１１０上のＫＶアドレスを格納する。

【００４５】

また順序キー列アクセス部１０４は、順序キー列の構造に従って、指定されたキーのエントリを順序キー列に追加する。キーに関連づけて、当該ＫＶアドレス割当部１０５が割り当てたＫＶアドレスを格納してもよい。このＫＶアドレスは、ハッシュテーブルに生成したエントリに設定したＫＶアドレスと同じである。

【００４６】

なお、ＫＶアドレス割当部１０５は、ハッシュテーブルアクセス部１０３または順序キー列アクセス部１０４の少なくとも一方の要求に応じて、ＫＶアドレスを割り当ててもよい。あるいは、処理振分部１０２からＫＶアドレスの割り当て要求を受けて、指定されたバリューを含むデータを格納するＫＶ記憶部１１０上のＫＶアドレスを割り当ててもよい。

【００４７】

ＫＶアクセス部１０６は、ＫＶアドレス割当部１０５で割り当てられたＫＶ記憶部１１０上のＫＶアドレスに、指定されたバリューを含むデータ（ここではキーバリュー）の書き込みを行う。

【００４８】

応答生成部１０７は、ＫＶアクセス部１０６による書き込みが完了すると、ＰＵＴ要求を実行できた旨の応答を生成して、ＰＵＴ要求の送信元に送信する。

【００４９】

ここで、指定されたキーと同じキーに対応するキーバリューが既にＫＶ記憶部１１０さ

れているかどうかを事前に確認する必要がある場合は、当該キーにマッチするエントリがハッシュテーブルに無いことを確認できた場合、または、ＫＶアクセス部１０６でキーを読み出した時点で、マッチするエントリが無いことを確認できた場合に、ＫＶアドレス割当部１０５は、ＫＶ記憶部１１０上に新規に格納するデータ（キーまたはキーバリューなど）のＫＶアドレスの割り当てを行えばよい。指定されたキーに対応するデータが既に保持されていた場合は、既にハッシュテーブルと順序キー列に該当エントリがあることになるので、それぞれに既に存在しているエントリのＫＶアドレスを、新しく割り当てられたＫＶアドレスで上書きすればよい。

【００５０】

なお、指定されたキーと同じキーに対応するデータ（バリューまたはキーバリューなど）を既に登録しているかどうかの確認は、上述ではハッシュテーブルアクセス部１０３経由で行う例を示したが、順序キー列アクセス部１０４を用いてもよい。すなわち、順序キー列に、指定されたキーと同じキーが存在するかを確認すればよい。なお、ハッシュテーブルアクセス部１０３経由で行った方が、処理が高速になるという利点がある。

【００５１】

図５に、本実施形態の動作例のフローチャートを示す。ここでは読み出しに係る要求として、ＧＥＴＮＥＸＴ要求を受け付けた場合の動作例を示す。

【００５２】

まず、要求解釈部１０１がクライアント装置（図１参照）から、ネットワークを介してＧＥＴＮＥＸＴ要求を受け付ける（Ｓ１０１）。ＧＥＴＮＥＸＴ要求は、インターネットであればＴＣＰ／ＩＰパケット等により運ばれる。

【００５３】

処理振分部１０２は、ＧＥＴＮＥＸＴ要求で指定されたキーと、要求の処理内容とを順序キー列アクセス部１０４に通知する。順序キー列アクセス部１０４は、順序キー列に基づき、指定されたキーの次のキー（指定されたキーに関連するキー）を特定し、処理振分部１０２に渡す（Ｓ１０２）。

【００５４】

処理振分部１０２は、当該キーをハッシュテーブルアクセス部１０３に渡し、ハッシュテーブルアクセス部１０３は、当該キーのハッシュ値を計算する。そして、ハッシュテーブルにおいて当該ハッシュ値に基づくハッシュアドレスに対応するエントリに格納されているＫＶアドレスを取得する（Ｓ１０３）。

【００５５】

ＫＶアクセス部１０６は、ＫＶ記憶部１１０において当該ＫＶアドレスに格納されているデータ（キーバリューなど）を読み出し、当該データからバリューを取り出す（Ｓ１０４）。

【００５６】

応答生成部１０７は、ＫＶアクセス部１０６により取得されたバリューを含む応答を生成し、当該応答をＧＥＴＮＥＸＴ要求の送信元に送信する（Ｓ１０５）。

【００５７】

ここでは、ＧＥＴＮＥＸＴ要求を受け受けた場合の動作を説明したが、ＧＥＴＰＲＥＶ要求、または、ＧＥＴＲＡＮＧＥ要求などを受けた場合も同様にして行われる。

【００５８】

図６に、本実施形態の他の動作例のフローチャートを示す。ここでは、書き込みに係る要求として、ＰＵＴ要求を受け付けた場合の動作例を示す。

【００５９】

まず、要求解釈部１０１がクライアント装置から、ネットワークを介してＰＵＴ要求を受け付ける（Ｓ２０１）。

【００６０】

処理振分部１０２は、ＰＵＴ要求で指定されたキーとバリューとをハッシュテーブルアクセス部１０３に渡し、ハッシュテーブルアクセス部１０３は、当該キーのハッシュ値を

計算する。またＫＶアドレス割当部１０５にＫＶアドレスの割り当てを要求してＫＶアドレスの割り当てを受け（Ｓ２０２）、ハッシュテーブルにおいて当該ハッシュ値に基づくハッシュアドレスに対応するエントリにＫＶアドレスを格納する（Ｓ２０３）。

【００６１】

処理振分部１０２は、ＰＵＴ要求で指定されたキーを順序キー列アクセス部１０４に渡す。順序キー列アクセス部１０４は、順序キー列に、指定されたキーを追加する（Ｓ２０４）。

【００６２】

ＫＶアクセス部１０６は、ＫＶアドレス割当部１０５で割り当てられたＫＶ記憶部１１０上のＫＶアドレスに、指定されたバリューを含むデータ（キーバリューなど）の書き込みを行う（Ｓ２０５）。

【００６３】

応答生成部１０７は、ＫＶアクセス部１０６による書き込みが完了すると、ＰＵＴ要求を実行できた旨の応答を生成して、ＰＵＴ要求の送信元に送信する（Ｓ２０６）。

【００６４】

以上、本実施形態によれば、ＧＥＴ要求に対してはハッシュテーブル方式並みの高速な検索を行って応答を返しつつ、ＧＥＴＮＥＸＴ要求／ＧＥＴＰＲＥＶ要求／ＧＥＴＲＡＮＧＥ要求などのキー順序を考慮する必要がある要求にも、高速に検索して応答することが可能になる。

【００６５】

（第２の実施形態）

図７に、本発明の第２の実施形態に係るデータ処理装置を示す。図７のデータ処理装置は、第１の実施形態のデータ処理装置を拡張したものである。第１の実施形態と異なり、内部キー生成部１１１が追加され、順序キー列記憶部１０９が除去されている。第２の実施形態では、順序キー列をバリューとして、ＫＶ記憶部１１０で管理するところが異なる。以下、第１の実施形態との差分を中心に説明し、重複する構成および動作の説明は適宜省略する。

【００６６】

第１の実施形態では、順序キー列は順序キー列記憶部１０９（図２参照）で管理され、一例としてファイルシステムで管理されることを述べた。順序キー列は、全てのキーを辞書順に並べて記憶しているので、それだけで容量が大きくなるため、一般に複数の部分に分割されている。個々の部分は全体の部分集合となり、それ自体でキーを辞書順に並べて管理している。当然ながら既存のキーの並び間に、新たなキーを挿入したり、あるいはキーの並びの中の途中のキーを削除したりする必要がある。このことから、キーの位置や長さが柔軟に変更可能である必要があり、また、部分自体を新規に生成したり削除したりすることも容易にできる必要がある。これには上述したファイルシステムを利用することで対応でき、個々の部分は個々のファイルとして実装される。これにより実装が容易となるが、一方でファイルシステムを用いると、汎用で複雑な機構を設ける必要があり、構成全体として複雑になり、性能的にもあまり効率が良くなくなる可能性がある。

【００６７】

そこで本実施形態では、ファイルシステムを使わずに、順序キー列を、内部的なキーバリューとしてＫＶ記憶部１１０で管理する構成をとる。

【００６８】

順序キー列アクセス部１０４は、ＰＵＴ要求を受けた場合、ＰＵＴ要求で指定されたキーを追加した新たな順序キー列を生成（詳細は後述）する。内部キー生成部１１１は、予め定めた方法で内部キーを生成する。一例として内部キーとして、常に同じ値を生成することがある。以下の説明では、内部キー生成部１１１は、常に同じ値の内部キーを生成するとする。処理振分部１０２は、内部キーと新たな順序キー列とをハッシュテーブルアクセス部１０３に渡す。ハッシュテーブルアクセス部１０３は、内部キーからハッシュ値を計算し、ハッシュテーブルに当該ハッシュ値に基づくハッシュアドレスに対応するエント

リを生成し、生成したエントリに、新たな順序キー列を格納するためのＫＶ記憶部１１０上のＫＶアドレスを格納する。当該ＫＶアドレスは、ＫＶアドレス割当部１０５に要求するなどして割り当てを受ければよい。ＫＶアクセス部１０６は、新たな順序キー列を、通常のバリュー（データ）として、ＫＶ記憶部１１０上のＫＶアドレスに格納する。

【００６９】

具体例として、例えばデータ処理装置を立ち上げて、最初のＰＵＴ（Ｋｅｙ＝“Ｊｏｈｎ”、Ｖａｌｕｅ＝“ＧＨＩ”）要求を受けたとする。このとき、指定されたキーとバリューが初めて、この装置に格納されることになるので、順序キー列としては、この指定されたキー（Ｊｏｈｎ）のみが存在することになる。よって、このキーを最初の順序キー列として、ＫＶ記憶部１１０に、内部キーに対応するデータ（バリュー）として格納するべく、例えば、内部キー生成部１１１は、Ｋｅｙ＝“ｋｅｙ＿ｓｅｑ０”を生成し、Ｖａｌｕｅ＝“Ｊｏｈｎ”とする。これらを処理振分部１０２に渡す。処理振分部１０２は、内部キーと“Ｊｏｈｎ”とをハッシュテーブルアクセス部１０３に渡す。ハッシュテーブルアクセス部１０３は、この内部キー “ｋｅｙ＿ｓｅｑ０”のハッシュ値を計算し、ハッシュテーブルにおける当該ハッシュ値のエントリに、“Ｊｏｈｎ”が格納されるＫＶアドレスを格納する。ＫＶアドレスは、ＫＶアドレス割当部１０５が割り当てる。ＫＶアクセス部１０６は、“Ｊｏｈｎ”を、通常のバリューとしてＫＶ記憶部１１０に格納する。なお、ＰＵＴ要求そのものに対する処理も、第１の実施形態と同様にして行う。すなわち、“Ｊｏｈｎ”のハッシュ値に対応するエントリに、“ＧＨＩ”を格納するＫＶ記憶部１１０上のＫＶアドレスを格納し、ＫＶ記憶部１１０上の当該ＫＶアドレスには、“ＧＨＩ”を含むデータを格納する。どちらの処理を先にするかは任意である。

【００７０】

さらに、ＰＵＴ（Ｋｅｙ＝“Ｂｏｂ”、Ｖａｌｕｅ＝“ＪＫＬ”）要求を受けた場合、まず順序キー列アクセス部１０４は、現在の順序キー列の読み出しを行う。すなわち、内部キー生成部１１１に内部キーの生成を依頼する。内部キー生成部１１１は、Ｋｅｙ＝“ｋｅｙ＿ｓｅｑ０”を生成し、処理振分部１０２に渡す。処理振分部１０２は、これをハッシュテーブルアクセス部１０３に送る。ハッシュテーブルアクセス部１０３は、“ｋｅｙ＿ｓｅｑ０”のハッシュ値を計算し、ハッシュ値に対応するエントリに格納されたＫＶアドレスを取得する。ＫＶアクセス部１０６は、ＫＶ記憶部１１０上の当該ＫＶアドレスからデータ、すなわち、内部キー “ｋｅｙ＿ｓｅｑ０”のバリューである“Ｊｏｈｎ”を読み出す。ＫＶアクセス部１０６は、これを順序キー列アクセス部１０４に渡す。これにより、順序キー列アクセス部１０４は、現在の順序キー列“Ｊｏｈｎ”を得る。そして、今回のＰＵＴ要求で指定されたＫｅｙ＝“Ｂｏｂ”と、現在の順序キー列“Ｊｏｈｎ”を比較し、“Ｂｏｂ”を“Ｊｏｈｎ”の前に配置することにより、新たな順序キー列として“Ｂｏｂ　Ｊｏｈｎ”を生成する。これを新たなバリューとし、また内部キーは固定の値である “ｋｅｙ＿ｓｅｑ０”とする。そして、前回同様に、ハッシュテーブルに、当該内部キーのハッシュ値に対応するエントリに、“Ｂｏｂ　Ｊｏｈｎ”を格納するためのＫＶ記憶部１１０上のＫＶアドレスを上書きし、“Ｂｏｂ　Ｊｏｈｎ”をＫＶ記憶部１１０上のＫＶアドレスに格納する。当該アドレスはＫＶアドレス割当部１０５により割り当てを受ける。なお、ＰＵＴ要求そのものに対する処理も行う。すなわち、“Ｂｏｂ”のハッシュ値に対応するエントリに、“ＪＫＬ”を格納するＫＶ記憶部１１０上のＫＶアドレスを格納し、ＫＶ記憶部１１０上の当該ＫＶアドレスには、“ＪＫＬ”を含むデータを格納する。どちらの処理を先にするかは任意である。

【００７１】

この後、さらにＰＵＴ（Ｋｅｙ＝“Ｋｅｎ”、Ｖａｌｕｅ＝“ＭＮＯ”）要求が来た場合、同様に、内部キーを “ｋｅｙ＿ｓｅｑ０”として、ＫＶ記憶部１１０からバリュー（データ）“Ｂｏｂ　Ｊｏｈｎ　Ｋｅｎ”を現在の順序キー列として読み出す。これに“Ｋｅｎ”を追加して、新たな順序キー列“Ｂｏｂ　Ｊｏｈｎ　Ｋｅｎ”を得る。ハッシュテーブルおよびＫＶ記憶部１１０を前回と同様にして更新する。なお、ＰＵＴ要求そのものに対する処理も前回と同様にして行う。

【００７２】
さらに、ＰＵＴ（Ｋｅｙ＝“Ｂｒｕｃｅ”、Ｖａｌｕｅ＝“ＰＱＲ”）要求が来た場合、同様に、内部キーを“ｋｅｙ＿ｓｅｑ０”として生成し、ＫＶ記憶部１１０からバリュー（データ）“Ｂｏｂ　Ｊｏｈｎ　Ｋｅｎ”を現在の順序キー列として読み出す。これに“Ｂｒｕｃｅ”を追加して、新たな順序キー列“Ｂｏｂ　Ｂｒｕｃｅ　Ｊｏｈｎ　Ｋｅｎ”を得る。ハッシュテーブルおよびＫＶ記憶部１１０を前回と同様にして更新する。なお、ＰＵＴ要求そのものに対する処理も前回と同様にして行う。
【００７３】
本実施形態におけるＫＶ記憶部１１０のデータ構造を図８に示す。順序キー列を含むデータ（バリュー）が、通常のキーに対応するバリューと同じ位置づけで、ＫＶ記憶部１１０に記憶されている。
【００７４】
次に、ＧＥＴＮＥＸＴ（Ｋｅｙ＝“Ｊｏｈｎ”）要求を受けた場合、順序キー列アクセス部１０４は、前述同様に、まず現在の順序キー列を読み出す。すなわち、内部キー生成部１１１に内部キーの生成を要求し、内部キー生成部１１１は、内部キーとして固定の値である“ｋｅｙ＿ｓｅｑ０”を生成して、処理振分部１０２に渡す。処理振分部１０２は、これをハッシュテーブルアクセス部１０３に渡して、内部キーに対応するバリュー（データ）が格納されているＫＶアドレスをハッシュテーブルから取得する。ＫＶアクセス部１０６は、ＫＶ記憶部１１０の当該取得したＫＶアドレスから、内部キー“ｋｅｙ＿ｓｅｑ０”のバリューである“Ｂｏｂ　Ｂｒｕｃｅ　Ｊｏｈｎ　Ｋｅｎ”を読み出す。これを順序キー列アクセス部１０４に渡し、これにより順序キー列アクセス部１０４は現在の順序キー列を得る。この順序キー列の情報から、ＧＥＴＮＥＸＴで指定されたキーであるＪｏｈｎの次のキーはＫｅｎであることを確認する。キー“Ｋｅｎ”を処理振分部１０２に渡し、処理振分部１０２は、これをハッシュテーブルアクセス部１０３に渡す。ハッシュテーブルアクセス部１０３は、“Ｋｅｎ”のハッシュ値に対応するエントリからＫＶアドレスを取得し、ＫＶアクセス部１０６が、ＫＶ記憶部１１０における当該ＫＶアドレスからデータを読み出し、読み出したデータから、キー“Ｋｅｎ”に対応するバリューを取り出す。応答生成部１０７が、取り出したバリューを含む応答を生成して、ＧＥＴＮＥＸＴ要求の送信元に送信する。
【００７５】
なお、順序キー列はＰＵＴ要求が来る毎にサイズが増大していくので、適宜分割して、それぞれに異なる内部キーを割り当ててもよい。例えば内部キーｋｅｙ＿ｓｅｑ０に対応する順序キー列を２つに分割し、それぞれ、内部キーとしてｋｅｙ＿ｓｅｑ１、ｋｅｙ＿ｓｅｑ２を割り当ててもよい。分割して管理するアルゴリズムは、ＬＳＭ－ｔｒｅｅなど従来の様々な方式に従ってよく、本実施形態はその内容を問わない。この場合、内部キー生成部１１１は、ＧＥＴ要求等の読み出しの要求を受けた場合は、当該アルゴリズムに基づき内部キーを特定すればよい。単純には、分割の際、アルファベットの範囲で分けてもよい。例えばＡ～Ｌ、Ｍ～Ｚの２つのグループに分けて、内部キーとしてｋｅｙ＿ｓｅｑ１、ｋｅｙ＿ｓｅｑ２を割り当てる。そして、読み出し要求で指定されたキーの先頭のアルファベットがＡ～Ｌのときはｋｅｙ＿ｓｅｑ１、Ｍ～Ｚのときはｋｅｙ＿ｓｅｑ２と決定してもよい。ここで述べた例は実現の可能性を示すための例に過ぎず、他にも様々な方法で実施可能である。
【００７６】
また、内部キー生成部１１１が生成する内部キーは、ＧＥＴ要求やＰＵＴ要求など、要求解析部で受け付ける要求で指定されるキー（外部キーと呼ぶ）と同じ値になる可能性もある。この問題を解消するため、外部キーと内部キーを明確に区別する情報を別途、外部キーおよび内部キーに付加してもよい。例えば、外部キーには例えば常に先頭に０を付け、内部キーには常に先頭に１を付ける。すなわち、外部キーが“Ｊｏｈｎ”であれば、先頭に０を付けて“０Ｊｏｈｎ”とし、内部キーが“ｋｅｙ＿ｓｅｑ０”であれば、先頭に１を付けて、“１ｋｅｙ＿ｓｅｑ０”とする。このように、内部キーおよび外部キー

に、特定の識別子を付けることで、内部キーと外部キーとを明確に区別できる。識別子の形式や付け方はどのような形態でもよい。

【００７７】

図９に、本実施形態の動作例のフローチャートを示す。ここでは、読み出しに係る要求として、ＧＥＴＮＥＸＴ要求を受け付けた場合の動作例を示す。

【００７８】

まず、要求解釈部１０１がクライアント装置から、ネットワークを介してＧＥＴＮＥＸＴ要求を受け付ける（Ｓ３０１）。

【００７９】

処理振分部１０２は、ＧＥＴＮＥＸＴ要求で指定されたキーを順序キー列アクセス部１０４に渡す。順序キー列アクセス部１０４は、内部キー生成部１１１に内部キーの生成を依頼し、内部キー生成部１１１は、内部キーを生成して、処理振分部１０２に送る（Ｓ３０２）。

【００８０】

処理振分部１０２は、内部キーをハッシュテーブルアクセス部１０３に渡し、ハッシュテーブルアクセス部１０３は、当該内部キーのハッシュ値を計算する。そして、ハッシュテーブルにおいて当該ハッシュ値に基づくハッシュアドレスに対応するエントリに格納されているＫＶアドレスを取得する（Ｓ３０３）。

【００８１】

ＫＶアクセス部１０６は、ＫＶ記憶部１１０において当該ＫＶアドレスに格納されているデータを読み出し、当該データからバリューである順序キー列を取り出す（Ｓ３０４）。

【００８２】

順序キー列アクセス部１０４は、当該順序キー列において、指定されたキーの次のキーを特定し、特定したキーを処理振分部１０２に送る（Ｓ３０５）。

【００８３】

処理振分部１０２は、当該キーをハッシュテーブルアクセス部１０３に渡し、ハッシュテーブルアクセス部１０３は、当該キーのハッシュ値を計算する。そして、ハッシュテーブルにおいて当該ハッシュ値に基づくハッシュアドレスに対応するエントリに格納されているＫＶアドレスを取得する（Ｓ３０６）。

【００８４】

ＫＶアクセス部１０６は、ＫＶ記憶部１１０において当該ＫＶアドレスに格納されているデータを読み出し、当該データからバリューを取り出す（Ｓ３０７）。

【００８５】

応答生成部１０７は、ＫＶアクセス部１０６により取得されたバリューを含む応答を生成し、当該応答をＧＥＴＮＥＸＴ要求の送信元に送信する（Ｓ３０８）。

【００８６】

ここでは、ＧＥＴＮＥＸＴ要求を受け受けた場合の動作を説明したが、ＧＥＴＰＲＥＶ要求、または、ＧＥＴＲＡＮＧＥ要求などを受けた場合も同様にして行われる。

【００８７】

図１０に、本実施形態の他の動作例のフローチャートを示す。ここでは、書き込みに係る要求として、ＰＵＴ要求を受け付けた場合の動作例を示す。

【００８８】

まず、要求解釈部１０１がクライアント装置から、ネットワークを介してＰＵＴ要求を受け付ける（Ｓ４０１）。

【００８９】

処理振分部１０２は、ＰＵＴ要求で指定されたキーを順序キー列アクセス部１０４に渡す。順序キー列アクセス部１０４は、内部キー生成部１１１に内部キーの生成を依頼し、内部キー生成部１１１は、内部キーを生成して、処理振分部１０２に送る（Ｓ４０２）。

【００９０】

処理振分部１０２は、当該内部キーをハッシュテーブルアクセス部１０３に渡し、ハッシュテーブルアクセス部１０３は、当該内部キーのハッシュ値を計算する。ハッシュ値に対応するエントリからＫＶアドレスを取得し（Ｓ４０３）、ＫＶアクセス部１０６に渡す。ＫＶアクセス部１０６は、ＫＶ記憶部１１０における当該ＫＶアドレスからデータを読み出し、データから順序キー列を取得する（Ｓ４０４）。

【００９１】

順序キー列アクセス部１０４は、取得された順序キー列に、ＰＵＴ要求で指定されたキーを追加して更新し、更新後の順序キー列をハッシュテーブルアクセス部１０３に渡す（Ｓ４０５）。ハッシュテーブルアクセス部１０３は、ＫＶアドレス割当部１０５にＫＶアドレスの割り当てを要求して、ＫＶアドレスの割り当てを受ける（Ｓ４０６）。ハッシュテーブルにおいて当該更新後の順序キー列のハッシュ値に基づくハッシュアドレスに対応するエントリに、ＫＶアドレスを格納（上書き）する（Ｓ４０７）。ＫＶアクセス部１０６は、ＫＶ記憶部１１０における当該ＫＶアドレスに、更新後の順序キーを含むデータを書き込む（Ｓ４０８）。

【００９２】

処理振分部１０２は、ＰＵＴ要求で指定されたキーとバリューとをハッシュテーブルアクセス部１０３に渡し、ハッシュテーブルアクセス部１０３は、当該指定されたキーのハッシュ値を計算する。ハッシュテーブルアクセス部１０３は、ＫＶアドレス割当部１０５にＫＶアドレスの割り当てを要求してＫＶアドレスの割り当てを受け（Ｓ４０９）、ハッシュテーブルにおいて当該ハッシュ値に基づくハッシュアドレスに対応するエントリにＫＶアドレスを格納する（Ｓ４１０）。

【００９３】

ＫＶアクセス部１０６は、ＫＶ記憶部１１０上の当該ＫＶアドレスに、指定されたバリューを含むデータの書き込みを行う（Ｓ４１１）。

【００９４】

応答生成部１０７は、ＫＶアクセス部１０６による書き込みが完了すると、ＰＵＴ要求を実行できた旨の応答を生成して、ＰＵＴ要求の送信元に送信する（Ｓ４１２）。

【００９５】

以上、本実施形態によれば、順序キー列をハッシュテーブルベースのＫＶＳ上にバリューとして管理することができるので、順序キー列を管理するためにファイルシステムを使用する必要はなくなる。これにより、全体のデータ構造の簡略化を図ることができる。

【００９６】

尚、各実施形態のデータ処理装置は、例えば、汎用のコンピュータ装置を基本ハードウェアとして用いることでも実現可能である。すなわち、上記のコンピュータ装置に搭載されたプロセッサにプログラムを実行させることにより実現出来る。このとき、データ処理装置は、上記のプログラムをコンピュータ装置にあらかじめインストールすることで実現することや、各種の記憶媒体に記憶、あるいはネットワークを介して上記のプログラムを配布、このプログラムをコンピュータ装置に適宜インストールすることで実現が出来る。また、データ処理装置内の各記憶部は、上記のコンピュータ装置に内蔵あるいは外付けされたメモリ、ハードディスクもしくはＣＤ－Ｒ、ＣＤ－ＲＷ、ＤＶＤ－ＲＡＭ、ＤＶＤ－Ｒなどの記憶媒体などを適宜利用して実現することができる。

【００９７】

本発明は上記実施形態そのままに限定されず、実施段階では要旨を逸脱しない範囲で構成要素を変形し具体化出来る。上記実施形態に開示されている複数の構成要素の適宜な組み合わせで、種々の発明を形成できる。例えば、実施形態に示される全構成要素から幾つかの構成要素を削除出来、異なる実施形態に渡る要素を適宜組み合わせることも出来る。

【符号の説明】

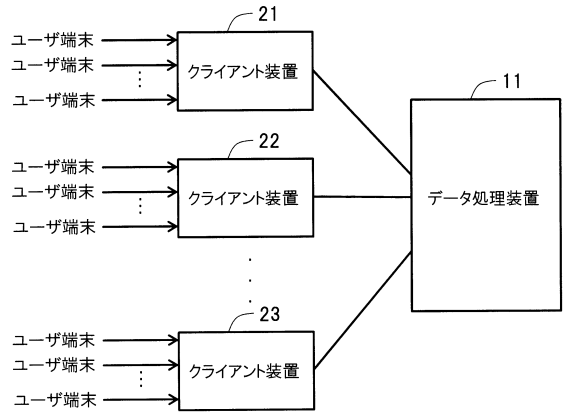
【００９８】

２１～２３：クライアント装置

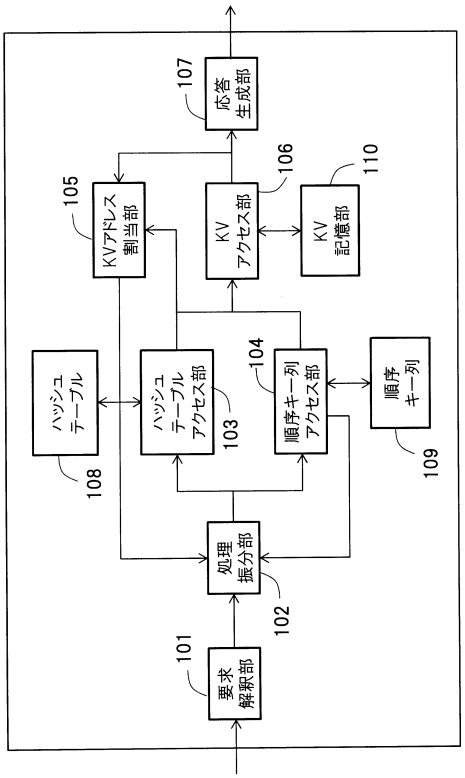
１１：データ処理装置

１０１：要求解釈部
１０２：処理振分部
１０３；ハッシュテーブルアクセス部
１０４：順序キー列アクセス部
１０５：ＫＶアドレス割当部
１０６：ＫＶアクセス部
１０７：応答生成部
１０８：ハッシュテーブル記憶部
１０９：順序キー列記憶部
１１０：ＫＶ記憶部
１１１：内部キー生成部

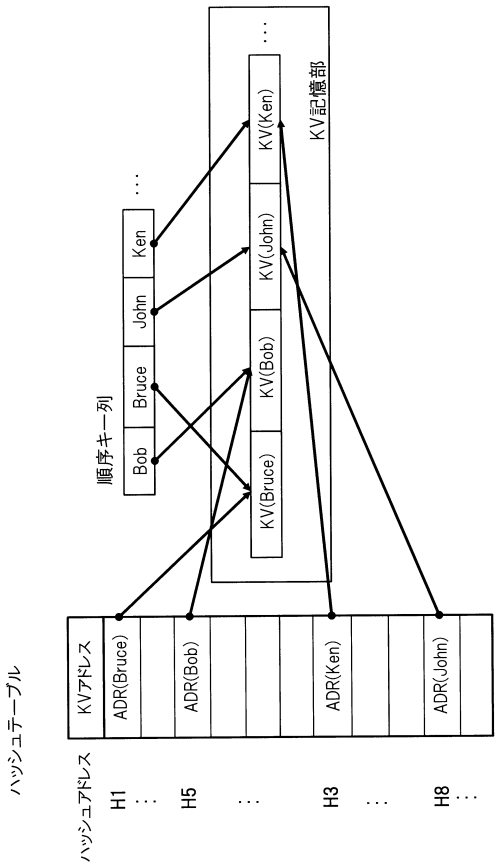
【図１】



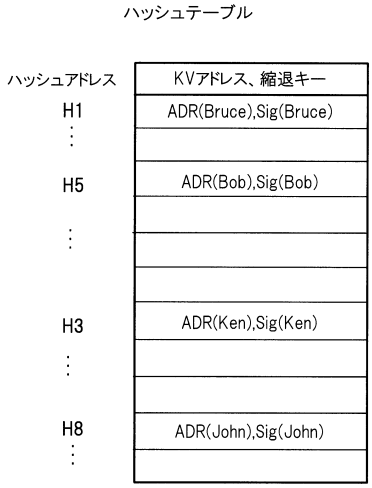
【図２】



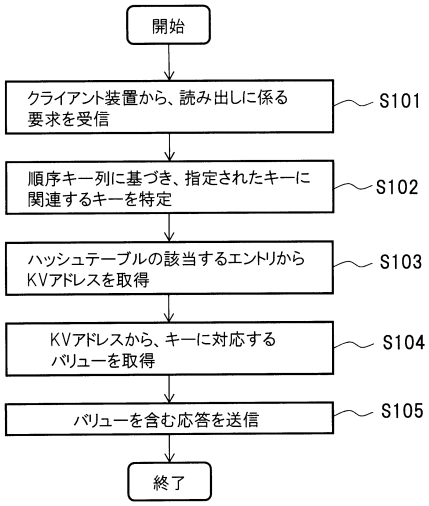
【図３】


KV記憶部
KV(Ken)
KV(John)
KV(Bob)
KV(Bruce)
順序キー列
Ken
John
Bruce
Bob
ハッシュテーブル
KVアドレス
ADR(Bruce)
ADR(Bob)
ADR(Ken)
ADR(John)
ハッシュアドレス
H1
H5
H3
H8

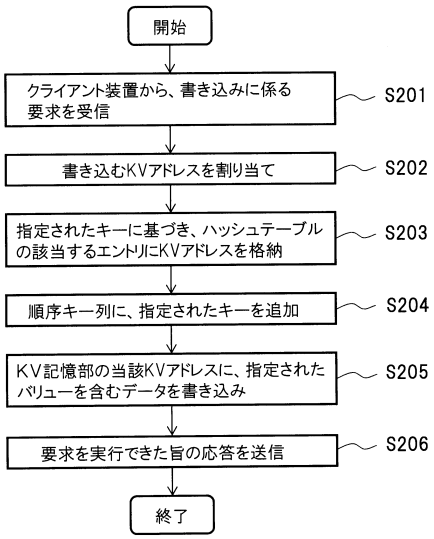
【図４】


ハッシュテーブル
ハッシュアドレス
KVアドレス、縮退キー
H1
ADR(Bruce),Sig(Bruce)
H5
ADR(Bob),Sig(Bob)
H3
ADR(Ken),Sig(Ken)
H8
ADR(John),Sig(John)

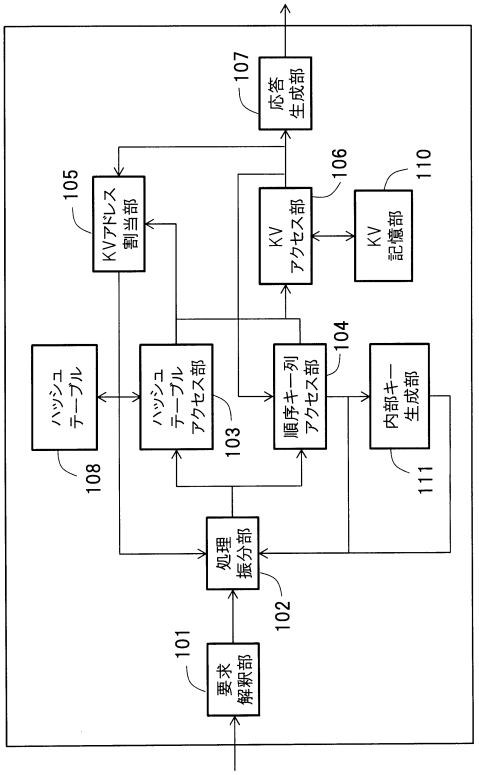
【図５】


開始
クライアント装置から、読み出しに係る要求を受信
S101
順序キー列に基づき、指定されたキーに関連するキーを特定
S102
ハッシュテーブルの該当するエントリからKVアドレスを取得
S103
KVアドレスから、キーに対応するバリューを取得
S104
バリューを含む応答を送信
S105
終了

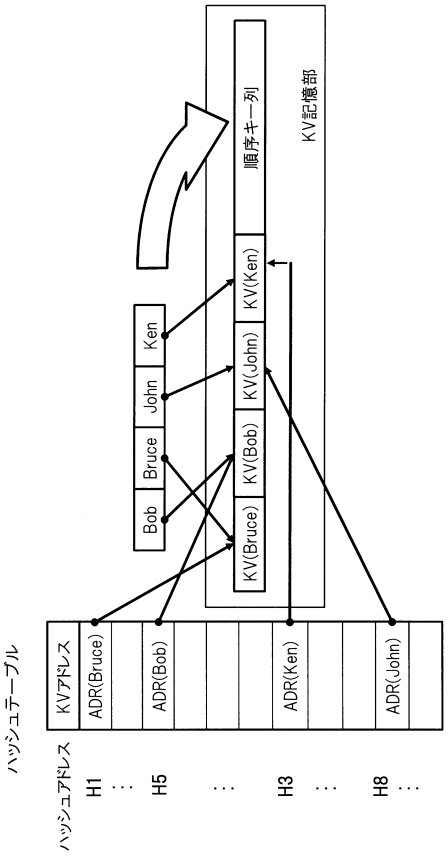
【図６】


開始
クライアント装置から、書き込みに係る要求を受信
S201
書き込むKVアドレスを割り当て
S202
指定されたキーに基づき、ハッシュテーブルの該当するエントリにKVアドレスを格納
S203
順序キー列に、指定されたキーを追加
S204
KV記憶部の当該KVアドレスに、指定されたバリューを含むデータを書き込み
S205
要求を実行できた旨の応答を送信
S206
終了

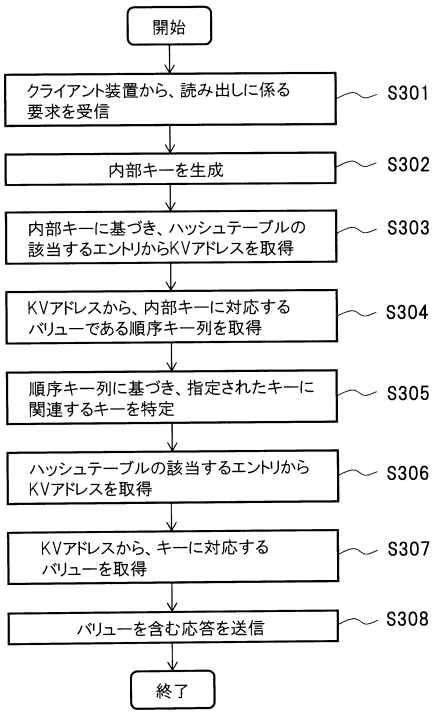
【図７】


107
応答生成部
105
KVアドレス割当部
106
KVアクセス部
110
KV記憶部
ハッシュテーブル
ハッシュテーブルアクセス部
順序キー列アクセス部
104
内部キー生成部
108
103
111
処理振分部
102
101
要求解釈部

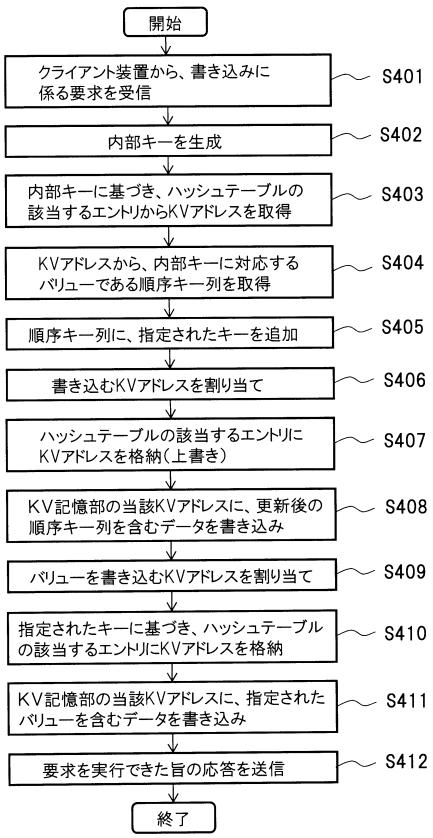
【図８】


KV記憶部
順序キー列
KV(Ken)
KV(John)
KV(Bob)
KV(Bruce)
Ken
John
Bruce
Bob
ハッシュテーブル
KVアドレス
ADR(Bruce)
ADR(Bob)
ADR(Ken)
ADR(John)
ハッシュアドレス
H1 … H5 … H3 … H8 …

【図９】


開始
クライアント装置から、読み出しに係る要求を受信
S301
内部キーを生成
S302
内部キーに基づき、ハッシュテーブルの該当するエントリからKVアドレスを取得
S303
KVアドレスから、内部キーに対応するバリューである順序キー列を取得
S304
順序キー列に基づき、指定されたキーに関連するキーを特定
S305
ハッシュテーブルの該当するエントリからKVアドレスを取得
S306
KVアドレスから、キーに対応するバリューを取得
S307
バリューを含む応答を送信
S308
終了

【図１０】


開始
クライアント装置から、書き込みに係る要求を受信
S401
内部キーを生成
S402
内部キーに基づき、ハッシュテーブルの該当するエントリからKVアドレスを取得
S403
KVアドレスから、内部キーに対応するバリューである順序キー列を取得
S404
順序キー列に、指定されたキーを追加
S405
書き込むKVアドレスを割り当て
S406
ハッシュテーブルの該当するエントリにKVアドレスを格納（上書き）
S407
KV記憶部の当該KVアドレスに、更新後の順序キー列を含むデータを書き込み
S408
バリューを書き込むKVアドレスを割り当て
S409
指定されたキーに基づき、ハッシュテーブルの該当するエントリにKVアドレスを格納
S410
KV記憶部の当該KVアドレスに、指定されたバリューを含むデータを書き込み
S411
要求を実行できた旨の応答を送信
S412
終了

フロントページの続き

(74)代理人 100118876
弁理士 鈴木 順生
(72)発明者 田中 信吾
東京都港区芝浦一丁目１番１号 株式会社東芝内

審査官 塚田 肇

(56)参考文献 特開２０１５－１７６４０７（ＪＰ，Ａ）
国際公開第２０１４／１４１５９４（WO，Ａ１）
特開２００９－００３５４１（ＪＰ，Ａ）
特開２００６－０５３７１１（ＪＰ，Ａ）

(58)調査した分野(Int.Cl.，ＤＢ名)
Ｇ０６Ｆ １２／００