

(19)日本国特許庁(JP)

(12)特許公報(B2)

(11)特許番号

特許第7095074号

(P7095074)

(45)発行日 令和4年7月4日(2022.7.4)

(24)登録日 令和4年6月24日(2022.6.24)

(51)国際特許分類

F I

G 0 6 F 12/123(2016.01)

G 0 6 F 12/123

G 0 6 F 3/06 (2006.01)

G 0 6 F 3/06 3 0 2 A

G 0 6 F 12/06 (2006.01)

G 0 6 F 12/06 5 1 5 J

G 0 6 F 12/0866(2016.01)

G 0 6 F 12/0866 1 0 0

請求項の数 13 (全21頁)

(21)出願番号 特願2020-507995(P2020-507995)

(86)(22)出願日 平成30年8月10日(2018.8.10)

(65)公表番号 特表2020-531959(P2020-531959
A)

(43)公表日 令和2年11月5日(2020.11.5)

(86)国際出願番号 PCT/IB2018/056028

(87)国際公開番号 WO2019/034976

(87)国際公開日 平成31年2月21日(2019.2.21)

審査請求日 令和3年1月22日(2021.1.22)

(31)優先権主張番号 15/680,577

(32)優先日 平成29年8月18日(2017.8.18)

(33)優先権主張国・地域又は機関
米国(US)

(73)特許権者 390009531

インターナショナル・ビジネス・マシー
ンズ・コーポレーションINTERNATIONAL BUSI
NESS MACHINES CORPO
RATIONアメリカ合衆国10504 ニューヨー
ク州 アーモンク ニュー オーチャード
ロードNew Orchard Road, A
rmonk, New York 105
04, United States of
America

(74)代理人 100112690

弁理士 太佐 種一

最終頁に続く

(54)【発明の名称】 キャッシュ管理のための方法、システム、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】

キャッシュ管理の方法であって、

キャッシュ中の第1のタイプのトラックのリストを生成するステップと、

前記キャッシュ中に第2のタイプのトラックのリストを生成するステップであって、I/Oオペレーションは、前記第2のタイプのトラックに対してよりも前記第1のタイプのトラックに対して相対的により速く完了される、前記生成するステップと、

ある1つのトラックを前記第1のタイプのトラックの前記リストから、または前記第2のタイプのトラックの前記リストからデモートすべきかどうかを判定するステップと

を備える、方法。

【請求項2】

前記第1のタイプのトラックは、syncioトラックであり、syncioトラックに対してI/Oオペレーションを行うために、アプリケーション・スレッドがスピン・ループ中に保持されて、前記I/Oオペレーションが完了するのを待機する、請求項1に記載の方法。

【請求項3】

前記第2のタイプのトラックは、non-syncioトラックであり、non-syncioトラックに対して前記I/Oオペレーションを行うために、前記アプリケーション・スレッドを前記スピン・ループ中に保持するステップが回避され、前記トラック上で行われる最後のI/Oオペレーションがsyncioオペレーションを備える第1のタイプ

の I / O オペレーションまたは `non - sync io` オペレーションを備える第 2 のタイプの I / O オペレーションであるかどうかに基づいて、ある 1 つのトラックが前記第 1 のタイプのトラックの前記リストへ、または前記第 2 のタイプのトラックの前記リストへ追加される、請求項 2 に記載の方法。

【請求項 4】

前記第 1 のタイプのトラックの前記リスト中のトラックは、前記第 1 のタイプのトラックの前記リスト中の各トラックがどれほど最近に用いられたかに基づいて順序付けられ、前記第 2 のタイプのトラックの前記リスト中のトラックは、前記第 2 のタイプのトラックの前記リスト中の各トラックがどれほど最近に用いられたかに基づいて順序付けられる、請求項 3 に記載の方法。

10

【請求項 5】

前記方法は、

前記第 2 のタイプのトラックの前記リスト中のトラックについてのリード・ヒット率が前記第 1 のタイプのトラックにおけるトラックについてのリード・ヒット率の所定の係数より小さいと判定するステップにตอบสนองして、ある 1 つのトラックを前記第 2 のタイプのトラックの前記リストからデモートするステップ
をさらに備える、請求項 4 に記載の方法。

【請求項 6】

前記第 1 のタイプのトラックの前記リスト中のトラックについての前記リード・ヒット率および前記第 2 のタイプのトラックの前記リスト中のトラックについての前記リード・ヒット率は、前記第 1 のタイプのトラックおよび前記第 2 のタイプのトラックの前記リストの所定の底の部分に対するリード・ヒットに基づいて計算される、請求項 5 に記載の方法。

20

【請求項 7】

前記方法は、

入力 / 出力 (I / O) オペレーションの速度を増加させるために前記所定の係数を調整するステップ
をさらに備える、請求項 5 に記載の方法。

【請求項 8】

キャッシュ管理のためのシステムであって、

メモリ、および

前記メモリに結合されたプロセッサを備え、前記プロセッサは、オペレーションを行い、前記プロセッサによって行われる前記オペレーションは、

キャッシュ中の第 1 のタイプのトラックのリストを生成するステップと、

前記キャッシュ中に第 2 のタイプのトラックのリストを生成するステップであって、 I / O オペレーションは、前記第 2 のタイプのトラックに対してよりも前記第 1 のタイプのトラックに対して相対的により速く完了される、前記生成するステップと、

ある 1 つのトラックを前記第 1 のタイプのトラックの前記リストから、または前記第 2 のタイプのトラックの前記リストからデモートすべきかどうかを判定するステップと

を備える、システム。

30

【請求項 9】

前記第 1 のタイプのトラックは、 `sync io` トラックであり、 `sync io` トラックに対して I / O オペレーションを行うために、アプリケーション・スレッドがスピン・ループ中に保持されて、前記 I / O オペレーションが完了するのを待機する、請求項 8 に記載のシステム。

40

【請求項 10】

前記第 2 のタイプのトラックは、 `non - sync io` トラックであり、 `non - sync io` トラックに対して前記 I / O オペレーションを行うために、前記アプリケーション・スレッドを前記スピン・ループ中に保持するステップが回避され、前記トラック上で行われる最後の I / O オペレーションが `sync io` オペレーションを備える第 1 のタイプの I / O オペレーションまたは `non - sync io` オペレーションを備える第 2 のタイ

50

プの I / O オペレーションであるかどうかに基づいて、ある 1 つのトラックが前記第 1 のタイプのトラックの前記リストへ、または前記第 2 のタイプのトラックの前記リストへ追加される、請求項 9 に記載のシステム。

【請求項 11】

前記第 1 のタイプのトラックの前記リスト中のトラックは、前記第 1 のタイプのトラックの前記リスト中の各トラックがどれほど最近に用いられたかに基づいて順序付けられ、前記第 2 のタイプのトラックの前記リスト中のトラックは、前記第 2 のタイプのトラックの前記リスト中の各トラックがどれほど最近に用いられたかに基づいて順序付けられる、請求項 10 に記載のシステム。

【請求項 12】

キャッシュ管理のためのコンピュータ・プログラムであって、
キャッシュ中の第 1 のタイプのトラックのリストを生成するステップと、
前記キャッシュ中に第 2 のタイプのトラックのリストを生成するステップであって、I / O オペレーションは、前記第 2 のタイプのトラックに対してよりも前記第 1 のタイプのトラックに対して相対的により速く完了される、前記生成するステップと、
ある 1 つのトラックを前記第 1 のタイプのトラックの前記リストから、または前記第 2 のタイプのトラックの前記リストからデモットすべきかどうかを判定するステップと
をコンピュータに実行させるコンピュータ・プログラム。

【請求項 13】

キャッシュ管理のためのストレージ・コントローラであって、
キャッシュ、および前記ストレージ・コントローラ中に維持されたキャッシュ管理アプリケーションを備え、前記キャッシュ管理アプリケーションは、オペレーションを行い、前記オペレーションは、
前記キャッシュ中の第 1 のタイプのトラックのリストを生成するステップと、
前記キャッシュ中に第 2 のタイプのトラックのリストを生成するステップであって、I / O オペレーションは、前記第 2 のタイプのトラックに対してよりも前記第 1 のタイプのトラックに対して相対的により速く完了される、前記生成するステップと、
ある 1 つのトラックを前記第 1 のタイプのトラックの前記リストから、または前記第 2 のタイプのトラックの前記リストからデモットすべきかどうかを判定するステップと
を備える、ストレージ・コントローラ。

【発明の詳細な説明】

【技術分野】

【0001】

本発明の複数の実施形態は、入力 / 出力 (I / O : i n p u t / o u t p u t) オペレーションのタイプに基づくキャッシュ管理に関する。

【背景技術】

【0002】

あるストレージ・システム環境において、ストレージ・コントローラ (またはストレージ・コントローラ・コンプレックス) は、互いに結合された複数のストレージ・サーバを備えてよい。ストレージ・コントローラは、ホスト・コンピューティング・システムが、ストレージ・コントローラによって制御されたストレージ・デバイスを用いて入力 / 出力 (I / O) オペレーションを行うことを許容してよく、ここで、ホスト・コンピューティング・システムは、ホストと呼ばれることがある。

【0003】

ストレージ・コントローラは、データに対するホストからの将来のリクエストがより速くサービスされるようにそのデータを記憶するキャッシュを含む。データは、ストレージ・デバイスへ書き込み、またはそれから読み取るのと比較してずっとより速くキャッシュへ書き込まれ、またはそれから読み取られうる。キャッシュ・ヒットは、ホストからリクエストされたデータがキャッシュ中で見つかるときに発生し、キャッシュ・ミスは、リクエストされたデータがキャッシュ中で見つからないときに発生する。キャッシュ・ヒットは

10

20

30

40

50

、キャッシュからデータを読み取ることによってサービスされ、この読み取りは、ストレージ・コントローラに結合されたストレージ・デバイスからの読み取りより速い。

【発明の概要】

【発明が解決しようとする課題】

【0004】

キャッシュは、スペースまたはセグメントが不足しているときには、データ項目をキャッシュからエビクト (evict) する必要がある、データ項目のキャッシュからのこのエビクション (eviction) がデモーションと呼ばれる。新しいデータ項目のための余地を作るのにキャッシュ中のどのデータ項目がデモートされるべきかを決定するためにキャッシュ置換ポリシーが用いられてよい。リースト・リーセントリ・ユーズド (LRU: least recently used) ポリシーは、最近、最も長い時間にわたってアクセスされなかった項目を最初にデモート (demote) する。本発明は、キャッシュ管理のための方法、システム、およびコンピュータ・プログラムを提供する。

10

【課題を解決するための手段】

【0005】

典型的な実施形態において、キャッシュ中の第1のタイプのトラックのリストが生成され、キャッシュ中の第2のタイプのトラックのリストが生成され、I/Oオペレーションは、第2のタイプのトラックに対してよりも第1のタイプのトラックに対して相対的により速く完了される。ある1つのトラックを第1のタイプのトラックのリストから、または第2のタイプのトラックのリストからデモートすべきかどうかについて判定が行われる。判定の結果として、デモートされるトラックがI/Oオペレーションの速度を増加させる。

20

【0006】

さらなる実施形態例において、第1のタイプのトラックは、syncioトラックであり、syncioトラックに対してI/Oオペレーションを行うためには、アプリケーション・スレッドがスピン・ループ中に保持されて、そのI/Oオペレーションが完了するのを待機する。結果として、ある実施形態においては、I/Oオペレーションの速度を増加させるために、syncioトラックをデモートしない方が望ましいことがある。

【0007】

なおさらなる実施形態において、第2のタイプのトラックは、non-syncioトラックであり、non-syncioトラックに対してI/Oオペレーションを行うためには、アプリケーション・スレッドをスピン・ループ中に保持するステップが回避され、トラック上で行われる最後のI/Oオペレーションがsyncioオペレーションを備える第1のタイプのI/Oオペレーションまたはnon-syncioオペレーションを備える第2のタイプのI/Oオペレーションであるかどうかに基づいて、ある1つのトラックが第1のタイプのトラックのリストへ、または第2のタイプのトラックのリストへ追加される。結果として、2つの異なるタイプのリストを構築することによってI/Oオペレーションの速度が増加される。

30

【0008】

さらなる実施形態において、第1のタイプのトラックのリスト中のトラックは、第1のタイプのトラックのリスト中の各トラックがどれほど最近に用いられたかに基づいて順序付けられ、第2のタイプのトラックのリスト中のトラックは、第2のタイプのトラックのリスト中の各トラックがどれほど最近に用いられたかに基づいて順序付けられる。結果として、2つの異なるタイプのトラックのリースト・リーセントリ・ユーズド・リストを構築することによってI/Oオペレーションの速度が増加される。

40

【0009】

ある実施形態において、第2のタイプのトラックのリスト中のトラックについてのリード・ヒット率が第1のタイプのトラックにおけるトラックについてのリード・ヒット率の所定の係数より小さいと判定するステップにตอบสนองして、ある1つのトラックが第2のタイプのトラックのリストからデモートされる。第2のタイプのトラックからのデモーション (demotion) の結果として、I/Oオペレーションの速度が増加される。

50

【 0 0 1 0 】

追加の実施形態において、第 1 のタイプのトラックのリスト中のトラックについてのリード・ヒット率および第 2 のタイプのトラックのリスト中のトラックについてのリード・ヒット率が第 1 のタイプのトラックおよび第 2 のタイプのトラックのリストの所定の底の部分に対するリード・ヒットに基づいて計算される。結果として、トラックのデモーションは、より最近に用いられたトラックのセットに関する統計データを収集するステップに基づく。

【 0 0 1 1 】

さらなる実施形態において、入力 / 出力 (I / O) オペレーションの速度を増加させるために所定の係数が調整される。結果として、システム中の I / O オペレーションの速度がかかるデモーションの結果として減少しないかぎり、可能ならばいつでも、`syncio` トラックの LRU リストからのトラックではなく、むしろ `non-syncio` トラックの LRU リストからのトラックがデモートされる。

【 0 0 1 2 】

次に、同様の参照番号が全体にわたって対応する部分を表す図面を参照する。

【図面の簡単な説明】

【 0 0 1 3 】

【図 1】ある実施形態に従って、`syncio` ベースの I / O オペレーションがホストからストレージ・コントローラに向かって発生する、1 つ以上のホストおよび 1 つ以上のストレージ・デバイスに結合されたストレージ・コントローラを備えるコンピューティング環境のブロック図を図示する。

【図 2】ある実施形態に従って、トラックのリスト・リーセントリ・ユーズド (LRU) リストがどのようにストレージ・コントローラ中に維持されるかを示すブロック図を図示する。

【図 3】ある実施形態に従って、トラックの `non-syncio` LRU リストからまたは `syncio` LRU リストからのデモーションを示す第 1 のフローチャートを図示する。

【図 4】ある実施形態に従って、トラックの `non-syncio` LRU リストからまたは `syncio` LRU リストからのデモーションを示す第 2 のフローチャートを図示する。

【図 5】ある実施形態に従って、リード・ヒット率がどのように計算されるかを示すフローチャートを図示する。

【図 6】ある実施形態に従って、トラックのデモーションについての所定の係数がどのように計算されるかを示すブロック図を図示する。

【図 7】ある実施形態に従って、2 つの異なるタイプのトラックのリストを生成するステップに基づくトラックのデモーションを示すフローチャートを図示する。

【図 8】ある実施形態に従って、クラウド・コンピューティング環境のブロック図を図示する。

【図 9】ある実施形態に従って、図 8 のクラウド・コンピューティング環境のさらなる詳細のブロック図を図示する。

【図 10】ある実施形態に従って、図 1 ~ 9 に記載されるような、ストレージ・コントローラまたはホスト中に含まれてよいいくつかの要素を示す計算システムのブロック図を図示する。

【発明を実施するための形態】

【 0 0 1 4 】

以下の記載においては、本明細書の一部を形成し、かついくつかの実施形態を図示する添付図面が参照される。他の実施形態が利用されてもよく、構造およびオペレーションの変更が行われてもよいことが理解される。

【 0 0 1 5 】

`syncio` (`sync I / O` と呼ばれる) は、計算デバイスのための装着型ハード

10

20

30

40

50

ウェアおよびプロトコルを備える。syncioは、超低レイテンシのランダム・リードおよび小ブロックのシーケンシャル・ライト用に設計される。計算デバイス間のsyncio接続は、ポイント・ツー・ポイントの光学的ペリフェラル・コンポーネント・インターコネクト・エクスプレス(PCIe: Peripheral Component Interconnect Express)インターフェースを通じてよい。syncioオペレーションは、syncioではI/Oオペレーションが完了するのを待機している間、ホスト計算デバイスがアプリケーション・スレッドをスピン・ループ中に保持してよいという点で従来型I/Oとは異なって振舞う。これは、プロセッサ・サイクルが従来型I/Oの2つのコンテキストのスイッチを行う必要性、I/Oスレッドをスリープ状態にして、その後、I/Oスレッドを再ディスパッチするためのオペレーションを行う必要性、およびI/O割り込みの必要性を回避する。

10

【0016】

syncioオペレーションの時間要件を満たすためには、計算デバイス中のコード・パスが極度に最適化される必要がある。キャッシュ・ミスのような、syncioオペレーションを遅延させるいずれかの状態がsyncioオペレーションを行うことができないことを示すステータスを生成して、syncioオペレーションが再試行されなければならないことがある。syncioオペレーションは、キャッシュ・ヒットがあるときにのみ首尾よく行われうるので、ある実施形態がsyncioオペレーションのためのキャッシュ・ヒットを増加させるために提供される。かかる実施形態において、syncioトラックおよびnon-syncioトラックに対して別個のリスト・リーセントリ・ユーズド(LRU)リストを維持し、別個のLRUリストに基づくデモーション・メカニズムを用いることによって、単一のLRUリストのみが維持される実施形態と比較して、syncioオペレーションがより速く完了される。ある実施形態においては、可能であればいつでも、システム中のI/Oオペレーションの速度がかかるデモーションの結果として減少しない限り、syncioトラックのLRUリストからのトラックではなく、むしろnon-syncioトラックのLRUリストからのトラックがデモートされる。

20

【0017】

例示的な実施形態

図1は、ある実施形態に従って、1つ以上のホスト104、106および1つ以上のストレージ・デバイス108、110に結合されたストレージ・コントローラ102を備えるコンピューティング環境100のブロック図を図示する。ストレージ・コントローラ102は、複数のホスト104、106がストレージ・コントローラ102により維持された論理ストレージを用いて入力/出力(I/O)オペレーションを行うことを許容する。論理ストレージに対応する物理ストレージは、ストレージ・デバイス108、110もしくはストレージ・コントローラ102のキャッシュ112(例えば、メモリ)またはその両方のうちの1つ以上に見出されてよい。

30

【0018】

ストレージ・コントローラ102およびホスト104、106は、当技術分野において現在知られているもの、例えば、パーソナル・コンピュータ、ワークステーション、サーバ、メインフレーム、ハンド・ヘルド・コンピュータ、パーム・トップ・コンピュータ、テレフォニ・デバイス、ネットワーク・アプライアンス、ブレード・コンピュータ、処理デバイスなどを含めて、任意の適切な計算デバイスを備えてよい。ストレージ・コントローラ102、ホスト104、106およびストレージ・デバイス108、110は、任意の適切なネットワーク114、例えば、ストレージ・エリア・ネットワーク、ワイド・エリア・ネットワーク、インターネット、イントラネット中の要素であってよい。ある実施形態において、ストレージ・コントローラ102、ホスト104、106およびストレージ・デバイス108、110は、コンピューティング環境100を備えるクラウド・コンピューティング環境中の要素であってよい。ストレージ・デバイス108、110は、ストレージ・ディスク、テープ・ドライブ、固体ストレージなどからなっており、ストレージ・コントローラ102によって制御されてよい。

40

50

【0019】

ある実施形態において、ホスト104、106は、バス・インターフェース（例えば、ポイント・ツー・ポイントの光学的PCIeインターフェース）116、118およびネットワーク・インターフェース120、122を介してストレージ・コントローラ102に結合されてよい。ホスト104、106からのsyncioオペレーションは、バス・インターフェース116、118を通じて行われてよい。ホスト104、106からの従来型I/Oオペレーション（すなわち、non-syncioオペレーション）は、ネットワーク・インターフェース120、122を通じて行われてよい。バス・インターフェース116、118は、ネットワーク・インターフェース120、122より速いI/O用アクセス・チャンネルを備えてよい。イーサネットを通じてのPCIeを、例えば、Exp Ether技術を用いて、可能にするために、分散型PCIeスイッチのような、PCIe拡張ケーブルまたはコンポーネントを含めて、バス・インターフェース116、118を拡張するための追加のバス・インターフェース技術が利用されてもよい。

10

【0020】

キャッシュ112は、ある実施形態において、1つ以上のランクに区分された読み出し/書き込みキャッシュを備えてよく、ここで各ランクは、1つ以上のストレージ・トラックを含んでよい。キャッシュ112は、当技術分野で知られているまたは将来に開発される任意の適切なキャッシュであってよい。いくつかの実施形態において、キャッシュ112は、揮発性メモリもしくは不揮発性メモリまたはその両方を用いて実装されてよい。キャッシュ112は、修正および未修正データの両方を記憶してよい。キャッシュ112は、syncioトラック124およびnon-syncioトラック126を含む複数のトラックにデータを記憶してよい。syncioトラック124は、ホスト104、106からストレージ・コントローラ102上で行われるsyncioオペレーション128を介して読み出され、かつ書き込まれてよく、non-syncioトラック126は、ホスト104、106からストレージ・コントローラ102上で行われるnon-syncioオペレーション130を介して読み出され、かつ書き込まれてよい。

20

キャッシュ管理アプリケーション132は、キャッシュ112中の未修正syncioトラックのsyncioリスト・リーセントリ・ユーズド(LRU)リスト134およびキャッシュ112中の未修正non-syncioトラックのnon-syncio LRUリスト136を生成し、ここで未修正syncioトラックおよび未修正non-syncioトラックは、キャッシュ112中の空間を開放するためのキャッシュ112からのデモーションの候補である。ある実施形態において、キャッシュ管理アプリケーション132は、ソフトウェア、ハードウェア、ファームウェアまたはそれらの組み合わせで実装されてよい。

30

【0021】

図2は、ある実施形態に従って、トラックのLRUリストがどのようにストレージ・コントローラ102中に維持されるかを示すブロック図200を図示する。

【0022】

non-syncio LRUリスト134は、それに対してnon-syncioオペレーションが行われてよいキャッシュ112中の未修正non-syncioトラックのリストであり、syncio LRUリスト136は、それに対してsyncioオペレーションが行われてよいキャッシュ112中の未修正syncioトラックのリストである。

40

【0023】

non-syncio LRUリスト134中のエントリは、モースト・リーセントリ・ユーズド(most recently used)202からリスト・リーセントリ・ユーズド204まで上から下へ配置された、キャッシュ112中の未修正non-syncioトラックである。non-syncio LRUリスト134に対するリード・ヒット率を計算するために、non-syncio LRUリスト134の底の部分206におけるいくつかの所定のエントリが用いられてよい。

50

`non-syncio` LRUリスト134に対するリード・ヒット率を計算するために底の部分206（例えば、`non-syncio` LRUリスト134中の1000個のリスト・リーセントリ・ユーズド`non-syncio`トラック）のみを選択することによって、キャッシュ112からデモットされる可能性のあるトラックについてリード・ヒット率が決定される。

【0024】

`syncio` LRUリスト136中のエントリは、モースト・リーセントリ・ユーズド208からリスト・リーセントリ・ユーズド210まで上から下へ配置された、キャッシュ112中の未修正`syncio`トラックである。`syncio` LRUリスト136に対するリード・ヒット率を計算するために、`syncio` LRUリスト136の底の部分212におけるいくつかの所定のエントリが用いられてよい。`syncio` LRUリスト136に対するリード・ヒット率を計算するために底の部分212（例えば、`syncio` LRUリスト136中の1000個のリスト・リーセントリ・ユーズド`syncio`トラック）のみを選択することによって、キャッシュ112からデモットされる可能性のあるトラックについてリード・ヒット率が決定される。

10

【0025】

図3は、ある実施形態に従って、トラックの`non-syncio` LRUリストからまたは`syncio` LRUリストからのデモンションを示す第1のフローチャート300を図示する。図3に示されるオペレーションは、ストレージ・コントローラ102中で実行するキャッシュ管理アプリケーション132によって行われてよい。

20

【0026】

制御は、キャッシュ管理アプリケーション132が`syncio` LRUリスト136を生成して維持するブロック302において開始する。キャッシュ管理アプリケーション132は、`syncio` LRUリスト136中のトラックについてリード・ヒット率を（ブロック304において）決定し、制御は、ブロック302へ戻る。

【0027】

ブロック302、304に示されるオペレーションと並行して、キャッシュ管理アプリケーション132は、（ブロック306において）`non-syncio` LRUリスト134を生成して維持する。キャッシュ管理アプリケーション132は、`non-syncio` LRUリスト134中のトラックについてリード・ヒット率を（ブロック308において）決定し、制御は、ブロック306へ戻る。

30

【0028】

ブロック302、304、306、308のオペレーションが行われている間に、キャッシュ管理アプリケーション132は、トラックをキャッシュ112からデモットするためのオペレーションを（ブロック310において）行うステップを継続するかまたは開始する。キャッシュ管理アプリケーション132は、`non-syncio` LRUリスト134中のトラックについてのリード・ヒット率が`syncio` LRUリスト136中のトラックについてのリード・ヒット率の所定の係数（例えば、倍数）より小さいかどうかを（ブロック312において）判定する。例えば、`non-syncio` LRUリスト134中のトラックについてのリード・ヒット率が0.2であり、`syncio` LRUリスト136中のトラックについてのリード・ヒット率が0.1であり、所定の係数が3であれば、所定の係数を乗じた`syncio` LRUリスト136中のトラックについてのリード・ヒット率は、0.3である。`non-syncio` LRUリスト134中のトラックについてのリード・ヒット率は、0.3より小さい0.2である。それゆえに、この例では、`non-syncio` LRUリスト134中のトラックについてのリード・ヒット率が`syncio` LRUリスト136中のトラックについてのリード・ヒット率の所定の係数より小さい。

40

【0029】

ブロック312において、`non-syncio` LRUリスト134中のトラックについてのリード・ヒット率が`syncio` LRUリスト136中のトラックについてのリ

50

ード・ヒット率の所定の係数より小さいと判定されれば(「Yes」分岐314)、そのときにはキャッシュ管理アプリケーション132がトラックをnon-syncio LRUリスト134から(ブロック316において)デモートし、syncioオペレーションの性能は、syncioトラックがデモートされる状況を超えて向上されうる。non-syncio LRUリスト134中のトラックは、syncio LRUリスト136中のトラックより十分に少なく用いられており、従って、non-syncio LRUリスト134からデモートするのが望ましい。

【0030】

ブロック312において、non-syncio LRUリスト134中のトラックについてのリード・ヒット率がsyncio LRUリスト136中のトラックについてのリード・ヒット率の所定の係数より小さくないと判定されれば(「No」分岐318)、そのときにはキャッシュ管理アプリケーション132がトラックをsyncio LRUリスト136から(ブロック320において)デモートする。non-syncio LRUリスト134中のトラックは、syncio LRUリスト136中のトラックより十分に多く用いられており、従って、syncio LRUリスト136からデモートするのが望ましい。syncio LRUリスト136中のトラックに対する比較的少ないリード・ヒットがあつてよく、従って、トラックをsyncio LRUリスト136からデモートするのが望ましい。

10

【0031】

それゆえに、図3は、syncioおよびnon-syncioトラックに対して異なるLRUリストを維持することによって、単一のLRUリストのみが維持される実施形態と比較して、syncioオペレーションがより速く実行されうる実施形態を記載する。

20

【0032】

図4は、ある実施形態に従って、トラックのnon-syncio LRUリスト134からまたはsyncio LRUリスト136からのデモーションを示す第2のフローチャートを図示する。図4に示されるオペレーションは、ストレージ・コントローラ102中で実行するキャッシュ管理アプリケーション132によって行われてよい。

【0033】

制御は、キャッシュ管理アプリケーション132がsyncio LRUリスト136が空かどうかを判定するブロック402において開始する。空であれば(「Yes」分岐404)、制御は、キャッシュ管理アプリケーション132がトラックをnon-syncio LRUリスト134からデモートするブロック406へ進む。そうでない場合には(「No」分岐408)、制御は、キャッシュ管理アプリケーション132がnon-syncio LRUリスト134が空かどうかを判定するブロック410へ進む。

30

【0034】

ブロック410において、キャッシュ管理アプリケーション132がnon-syncio LRUリスト134が空であると判定すれば(「Yes」分岐412)、そのときにはキャッシュ管理アプリケーション132がトラックをsyncio LRUリスト136からデモートする。そうでない場合には(「No」分岐416)、キャッシュ管理アプリケーション132がnon-syncio LRUリスト134の最も古いトラックがsyncio LRUリスト136の最も古いトラックより古いかどうかを判定するブロック418へ進み、ここでリストの最も古いトラックとは、リスト中の他のすべての最後の使用より前の時間に最後に用いられた(すなわち、リード・ヒットがトラックに対して発生した)トラックであり、2番目のトラックの最後の使用の前の時間に最後に用いられた1番目のトラックは、2番目のトラックより古い。

40

【0035】

ブロック418において、キャッシュ管理アプリケーション132がnon-syncio LRUリスト134中の最も古いトラックがsyncio LRUリスト136中の最も古いトラックより古いと判定すれば(「Yes」分岐420)、制御は、キャッシュ管理アプリケーション132がトラックをnon-syncio LRUリスト134から

50

デモートするブロック 422 へ進む。そうでない場合には (「No」分岐 424)、キャッシュ管理アプリケーション 132 が (a) syncio LRUリストの底の部分 212、および (b) non-syncio LRUリストの底の部分 206 に対するリード・ヒット率を (ブロック 426 において) 計算する。

【0036】

syncio LRUリストの底の部分 212 に対するリード・ヒット率は、SIO_BOTTOMHITSと呼ばれ、non-syncio LRUリストの底の部分 206 に対するリード・ヒット率は、NONSIO_BOTTOMHITSと呼ばれる。

【0037】

ブロック 426 から、制御は、キャッシュ管理アプリケーション 132 が NONSIO_BOTTOMHITS が所定の係数を乗じた SIO_BOTTOMHITS より小さいかどうかを判定するブロック 428 へ進む。小さければ (「Yes」分岐 430)、キャッシュ管理アプリケーション 132 がトラックを non-syncio LRUリスト 134 からデモートする。そうでない場合には (「No」分岐 434)、キャッシュ管理アプリケーション 132 がトラックを syncio LRUリスト 136 から (ブロック 436 において) デモートする。

【0038】

それゆえに、図 4 は、non-syncio LRUリスト 134 および syncio LRUリスト 136 中の相対的により古いトラックであるトラックに対するリード・ヒットの比較が行われるように、non-syncio LRUリスト 134 および syncio LRUリスト 136 の少なくとも底の部分 206、212 がリード・ヒットの比較のために用いられる、ある実施形態を図示する。トラックのデモーションは、これらの相対的により古いトラックからであり、従って、リード・ヒット率は、これらの相対的により古いトラックについて計算される。

【0039】

図 5 は、ある実施形態に従って、リード・ヒット率がどのように計算されるかを示すフローチャート 500 を図示する。図 5 に示されるオペレーションは、ストレージ・コントローラ 102 中で実行するキャッシュ管理アプリケーション 132 によって行われてよい。

【0040】

制御は、キャッシュ管理アプリケーション 132 が syncio LRUリストの底の部分 212 についてリード・ヒット・カウンタを、および non-syncio LRUリストの底の部分 206 についてリード・ヒット・カウンタをゼロにセットするブロック 501 において開始する。制御は、リード・オペレーションがキャッシュ 112 に対するリード・ヒットを生成するブロック 502 へ進み、制御は、並行してブロック 504 および 506 へ進む。ブロック 504 において、キャッシュ管理アプリケーション 132 は、リード・ヒットが syncio LRUリストの底の部分 212 に対するものかどうかを判定し、そうであれば (「Yes」分岐 508)、syncio LRUリストの底の部分 212 についてリード・ヒットが (ブロック 510 において) インクリメントされる。そうでない場合には (「No」分岐 512)、制御は、ブロック 502 へ戻る。

【0041】

ブロック 506 において、キャッシュ管理アプリケーション 132 は、リード・ヒットが non-syncio LRUリストの底の部分 206 に対するものかどうかを判定し、そうであれば (「Yes」分岐 514)、non-syncio LRUリストの底の部分 206 についてリード・ヒット・カウンタが (ブロック 516 において) インクリメントされる。そうでない場合には (「No」分岐 518)、制御は、ブロック 502 へ戻る。ブロック 501、502、504、506、508、510、512、514、516、518 のオペレーションが (参照数字 520 によって示されるように) ある時間の間隔 (例えば、800 ミリ秒) にわたって繰り返して行われる。

【0042】

参照数字 520 によって示されるオペレーションが行われている間に、ブロック 522 お

10

20

30

40

50

よび524に示されるオペレーションが行われる。ブロック522において、キャッシュ管理アプリケーション132は、最後のN個の間隔についてリード・ヒットを加算して、それらの間隔にわたる累計時間で除することによって、syncioリストの底の部分212についてリード・ヒット率を計算する。ブロック524において、キャッシュ管理アプリケーション132は、最後のN個の間隔についてリード・ヒットを加算し、それらの間隔にわたる累計時間で除することによって、non-syncioリストの底の部分206についてリード・ヒット率を計算する。ある実施形態において、Nは、30のような数であってよい。

【0043】

それゆえに、図5は、図4に示されるオペレーションに用いるべくnon-syncioLRUリスト134およびsyncioLRUリスト136の底の部分に対するリード・ヒット率を計算するためのある実施形態を図示する。

10

【0044】

図6は、ある実施形態に従って、トラックのデモーションについての所定の係数がどのように計算されるかを示すブロック図600を図示する。例示的な所定の係数は、図3のブロック312、および図4のブロック428に用いられることが示された。

【0045】

ある実施形態において、所定の係数は、(参照数字602によって示されるように)システムの性能ベンチマークに基づいてよい。他の実施形態では、所定の係数は、(参照数字604によって示されるように)non-syncioワークロードよりもsyncioワークロードについてより良好な性能を所望するカスタマによって構成されてよい。なおさらなる実施形態において、所定の係数は、(参照数字606によって示されるように)ストレージ・コントローラ102において発生しているI/Oオペレーションの速度に基づいて動的に変更されてよい。例えば、所定の係数を増加するにつれてI/Oオペレーションの速度が増加すれば、そのときにはキャッシュ管理アプリケーション132が所定の係数を増加させ続け、所定の係数を増加するにつれてI/Oオペレーションの速度が減少すれば、そのときにはキャッシュ管理アプリケーション132が所定の係数を減少させ続ける。

20

【0046】

図7は、ある実施形態に従って、2つの異なるタイプのトラックのリストを生成するステップに基づくトラックのデモーションを示すフローチャート700を図示する。図7に示されるオペレーションは、ストレージ・コントローラ102中で実行するキャッシュ管理アプリケーション132によって行われてよい。

30

【0047】

制御は、キャッシュ112中の第1のタイプのトラックのリスト(例えば、syncioLRUリスト136)が生成されるブロック702において開始する。制御は、キャッシュ112中の第2のタイプのトラックのリスト(non-syncioLRUリスト134)が生成されるブロック704へ進み、I/Oオペレーションは、第2のタイプのトラックに対してよりも第1のタイプのトラックに対して相対的により速く完了され、トラック上で行われる最後のI/Oオペレーションが第1のタイプのI/Oオペレーション(例えば、syncioオペレーション)または第2のタイプのI/Oオペレーション(例えば、non-syncioオペレーション)であるかどうかに基づいて、ある1つのトラックが第1のタイプのトラックのリストに、または第2のタイプのトラックのリストに追加される。例えば、ある1つのトラック上で行われる最後のI/Oオペレーションがsyncioオペレーションであれば、そのときにはそのトラックがsyncioLRUリスト136に追加され、ある1つのトラック上で行われる最後のI/Oオペレーションがnon-syncioオペレーションであれば、そのときにはそのトラックがnon-syncioLRUリスト134に追加される。ある1つのトラックを第1のタイプのトラックのリストから、または第2のタイプのトラックのリストからデモートすべきかどうかについて(ブロック706において)判定が行われる。第1のタイプのトラックは、syncioトラックであってよく、ここでsyncioトラックに対してI/Oオペレ

40

50

ーションを行うために、アプリケーション・スレッドがスピン・ループ中に保持されて、I/Oオペレーションが完了するのを待機する。第2のタイプのトラックは、non-syncioトラックであってよく、ここでnon-syncioトラックに対してI/Oオペレーションを行うために、アプリケーション・スレッドをスピン・ループ中に保持するステップは、回避される。

【0048】

ある実施形態において、第1のタイプのトラックのリスト中のトラックは、（例えば、参照数字208、210によって示されるように）第1のタイプのトラックのリスト中の各トラックがどれほど最近に用いられたかに基づいて順序付けられ、第2のタイプのトラックのリスト中のトラックは、（例えば、参照数字202、204によって示されるように）第2のタイプのトラックのリスト中の各トラックがどれほど最近に用いられたかに基づいて順序付けられ、ここでリード・ヒットがある1つのトラックに対して発生すれば、そのトラックは用いられたと言われる。

【0049】

ブロック706に示されるトラック・デモーションに関する判定を行うために、ある実施形態において、第2のタイプのトラックのリスト中のトラックについてのリード・ヒット率が第1のタイプのトラックにおけるトラックについてのリード・ヒット率の所定の係数より小さいと判定するステップに回答して、ある1つのトラックが第2のタイプのトラックのリストから（ブロック708において）デモートされる。第1のタイプのトラックのリスト中のトラックについてのリード・ヒット率および第2のタイプのトラックのリスト中のトラックについてのリード・ヒット率は、第1のタイプのトラックおよび第2のタイプのトラックのリストの所定の底の部分に対するリード・ヒットに基づいて計算される。入力/出力（I/O）オペレーションの速度を増加させるために所定の係数が調整される。

【0050】

それゆえに、図1～7は、syncioトラックおよびnon-syncioトラックに対して別個のLRUリストを維持し、別個のLRUリストに基づくデモーション・メカニズムを用いることによって、単一のLRUリストのみが維持される実施形態と比較して、syncioオペレーションがより速く完了される、ある実施形態を図示する。ある実施形態において、可能であればいつでも、システム中のI/Oオペレーションの速度がかかるデモーションの結果として減少しない限り、syncioトラックのLRUリスト（syncio LRUリスト136）からのトラックではなく、むしろnon-syncioトラックのLRUリスト（non-syncio LRUリスト134）からのトラックがデモートされる。

【0051】

クラウド・コンピューティング環境

クラウド・コンピューティングは、最小限の管理努力またはサービス・プロバイダとの相互作用によって迅速にプロビジョンし、かつ解除することができるコンフィギュラブルなコンピューティング・リソース（例えば、ネットワーク、サーバ、ストレージ、アプリケーション、およびサービス）の共有プールへの便利なオンデマンド・ネットワーク・アクセスを可能にするためのモデルである。

【0052】

次に、図8を参照すると、例示的なクラウド・コンピューティング環境50が描かれる。示されるように、クラウド・コンピューティング環境50は、クラウド・コンシューマによって用いられるローカル・コンピューティング・デバイス、例えば、携帯情報端末（PDA: personal digital assistant）または携帯電話54A、デスクトップ・コンピュータ54B、ラップトップ・コンピュータ54C、もしくは自動車コンピュータ・システム54Nまたはそれらのすべてなどが通信してよい1つ以上のクラウド・コンピューティング・ノード10を備える。ノード10は、互いに通信してよい。これらのノードは、先に記載されたようなプライベート、コミュニティ、パブリック、もしくはハイブリッド・クラウド、またはそれらの組み合わせなど、1つ以上のネットワ

10

20

30

40

50

ークに、物理的または仮想的にグループ分けされてよい（示されない）。これは、クラウド・コンシューマがそのためにリソースをローカル・コンピューティング・デバイス上に維持する必要のないフラストラクチャ、プラットフォームもしくはソフトウェア・アズ・サービスまたはそれらのすべてをクラウド・コンピューティング環境50が提供することを許容する。図8に示されるタイプのコンピューティング・デバイス54A～Nは、例示に過ぎないことが意図され、コンピューティング・ノード10およびクラウド・コンピューティング環境50は、任意のタイプのネットワークもしくはネットワークによってアドレス可能な接続またはその両方を通じて（例えば、webブラウザを用いて）、任意のタイプのコンピュータ化デバイスと通信できることが理解される。

【0053】

次に、図9を参照すると、クラウド・コンピューティング環境50（図8）によって提供される機能抽象化層のセットが示される。図9に示されるコンポーネント、層、および機能は、例示に過ぎないことが意図され、本発明の実施形態は、それらに限定されないことが予め理解されるべきである。

【0054】

ハードウェアおよびソフトウェア層60は、ハードウェアおよびソフトウェア・コンポーネントを含む。ハードウェア・コンポーネントの例は、メインフレーム、一例ではIBM z Seriesシステム、RISC（縮小命令セット・コンピュータ：Reduced Instruction Set Computer）アーキテクチャ・ベースのサーバ、一例ではIBM p Seriesシステム、IBM x Seriesシステム、IBM BladeCenterシステム、ストレージ・デバイス、ネットワークおよびネットワーク・コンポーネントを含む。ソフトウェア・コンポーネントの例は、ネットワーク・アプリケーション・サーバ・ソフトウェア、一例ではIBM WebSphereアプリケーション・サーバ・ソフトウェア、およびデータベース・ソフトウェア、一例ではIBM DB2データベース・ソフトウェアを含む。（IBM、z Series、p Series、x Series、BladeCenter、WebSphere、およびDB2は、世界中の多くの管轄区域において登録されたInternational Business Machines Corporationの商標である）

【0055】

仮想化層62は、抽象化層を提供し、抽象化層からは仮想エンティティの以下の例、すなわち、仮想サーバ、仮想ストレージ、仮想プライベート・ネットワークを含めて、仮想ネットワーク、仮想アプリケーションおよびオペレーティング・システム、ならびに仮想クライアントが提供されてよい。

【0056】

一例において、管理層64は、以下に記載される機能を提供してよい。リソース・プロビジョニングは、クラウド・コンピューティング環境内でタスクを行うために利用されるコンピューティング・リソースおよび他のリソースの動的な調達を提供する。計量及び料金設定は、クラウド・コンピューティング環境内でリソースが利用される際のコスト追跡、およびこれらのリソースの消費に対する課金または請求を提供する。一例において、これらのリソースは、アプリケーション・ソフトウェア・ライセンスを備えてよい。セキュリティは、クラウド・コンシューマおよびタスクについての識別情報検証、ならびにデータおよび他のリソースのための保護を提供する。ユーザ・ポータルは、コンシューマおよびシステム・アドミニストレータのためにクラウド・コンピューティング環境へのアクセスを提供する。サービス・レベル管理は、要求されるサービス・レベルが満たされるようなクラウド・コンピューティング・リソース割り当ておよび管理を提供する。サービス・レベル・アグリメント（SLA：Service Level Agreement）計画および履行は、それらに対する将来の要求がSLAに従って予期されるクラウド・コンピューティング・リソースのための事前配置およびその調達を提供する。

【0057】

ワークロード層66は、クラウド・コンピューティング環境がそのために利用されてよい

10

20

30

40

50

機能性の例を提供する。この層から提供されてよいワークロードおよび機能の例は、マッピングおよびナビゲーション、ソフトウェア開発およびライフサイクル管理、仮想教室教育配信、データ解析処理、取引処理、および図 1 ~ 8 に示されるようなデモ・オペレーション 6 8 の処理を含む。

【 0 0 5 8 】

追加の実施形態の詳細

記載されるオペレーションは、ソフトウェア、ファームウェア、ハードウェア、またはそれらの任意の組み合わせを作り出すために標準的なプログラミングもしくは工学技術またはその両方を用いた方法、装置またはコンピュータ・プログラム製品として実装されてよい。それに応じて、実施形態の態様は、完全にハードウェアの実施形態、完全にソフトウェアの実施形態（ファームウェア、常駐ソフトウェア、マイクロコードなどを含む）、あるいは、本明細書ではすべてが全般的に「回路」、「モジュール」または「システム」と呼ばれてよいソフトウェアおよびハードウェアの態様を組み合わせた実施形態の形をとってよい。さらにまた、実施形態の態様は、コンピュータ・プログラム製品の形をとってもよい。コンピュータ・プログラム製品は、本実施形態の態様をプロセッサに実施させるためのコンピュータ可読プログラム命令をその上に有する（1つまたは複数の）コンピュータ可読ストレージ媒体を含んでよい。

【 0 0 5 9 】

コンピュータ可読ストレージ媒体は、命令実行デバイスによる使用のための命令を保持し、記憶できる有形のデバイスとすることができる。コンピュータ可読ストレージ媒体は、例えば、以下には限定されないが、電子ストレージ・デバイス、磁気ストレージ・デバイス、光ストレージ・デバイス、電磁ストレージ・デバイス、半導体ストレージ・デバイス、または前述のものの任意の適切な組み合わせであってよい。コンピュータ可読ストレージ媒体のより具体的な例の非網羅的なリストは、ポータブル・コンピュータ・ディスクレット、ハード・ディスク、ランダム・アクセス・メモリ（RAM：random access memory）、リード・オンリ・メモリ（ROM：read-only memory）、消去可能プログラマブル・リード・オンリ・メモリ（EPROM：erasable programmable read-only memoryまたはフラッシュ・メモリ）、スタティック・ランダム・アクセス・メモリ（SRAM：static random access memory）、ポータブル・コンパクト・ディスク・リード・オンリ・メモリ（CD-ROM：portable compact disc read-only memory）、デジタル多用途ディスク（DVD：digital versatile disk）、メモリ・スティック、フレキシブルディスク、パンチ・カードまたはその上に記録された命令を有する溝中の隆起構造のような機械的に符号化されたデバイス、および前述のものの任意の適切な組み合わせを含む。コンピュータ可読ストレージ媒体は、本明細書では、それ自体が一時的な信号、例えば、電波、または他の自由に伝搬する電磁波、導波路もしくは他の送信媒体を通して伝搬する電磁波（例えば、光ファイバ・ケーブルを通過する光パルス）、あるいはワイヤを通して送信される電気信号などであると解釈されるべきではない。

【 0 0 6 0 】

本明細書に記載されるコンピュータ可読プログラム命令を、コンピュータ可読ストレージ媒体からそれぞれのコンピューティング／処理デバイスへ、あるいはネットワーク、例えば、インターネット、ローカル・エリア・ネットワーク、ワイド・エリア・ネットワークもしくは無線ネットワークまたはそれらのすべてを介して外部コンピュータまたは外部ストレージ・デバイスへダウンロードできる。ネットワークは、銅送信ケーブル、光送信ファイバ、無線送信、ルータ、ファイヤウォール、スイッチ、ゲートウェイ・コンピュータもしくはエッジ・サーバまたはそれらのすべてを備えてよい。各コンピューティング／処理デバイス中のネットワーク・アダプタ・カードまたはネットワーク・インターフェースは、コンピュータ可読プログラム命令をネットワークから受信して、それらのコンピュータ可読プログラム命令をそれぞれのコンピューティング／処理デバイス内のコンピュータ

10

20

30

40

50

可読ストレージ媒体に記憶するために転送する。

【0061】

本実施形態のオペレーションを実施するためのコンピュータ可読プログラム命令は、アセンブラ命令、命令セット・アーキテクチャ (ISA: instruction-set-architecture) 命令、マシン命令、マシン依存命令、マイクロコード、ファームウェア命令、状態設定データ、または、Smalltalk、C++ などのようなオブジェクト指向プログラミング言語、および「C」プログラミング言語または同様のプログラミング言語のような従来の手続き型プログラミング言語を含む、1つ以上のプログラミング言語の任意の組み合わせで書かれたソース・コードまたはオブジェクト・コードのいずれかであってよい。コンピュータ可読プログラム命令は、全体的にユーザのコンピュータ上で、部分的にユーザのコンピュータ上で、スタンド・アロン・ソフトウェア・パッケージとして、部分的にユーザのコンピュータ上かつ部分的にリモート・コンピュータ上で、あるいは全体的にリモート・コンピュータまたはサーバ上で実行してよい。後者のシナリオにおいては、リモート・コンピュータが、ローカル・エリア・ネットワーク (LAN) またはワイド・エリア・ネットワーク (WAN) を含む、任意のタイプのネットワークを通してユーザのコンピュータへ接続されてもよく、あるいは外部コンピュータへ接続が (例えば、インターネット・サービス・プロバイダを用い、インターネットを通して) 行われてもよい。いくつかの実施形態においては、本実施形態の態様を行うために、例えば、プログラマブル・ロジック回路素子、フィールド・プログラマブル・ゲート・アレイ (FPGA: field-programmable gate array)、またはプログラマブル・ロジック・アレイ (PLA: programmable logic arrays) を含む、電子回路素子が、コンピュータ可読プログラム命令の状態情報を利用し、電子回路素子をパーソナライズすることによって、コンピュータ可読プログラム命令を実行してよい。

【0062】

本実施形態の態様は、本発明の実施形態による方法、装置 (システム)、およびコンピュータ・プログラム製品のフローチャート説明図もしくはブロック図またはその両方を参照して本明細書に記載される。フローチャート説明図もしくはブロック図またはその両方の各ブロック、およびフローチャート説明図もしくはブロック図またはその両方におけるブロックの組み合わせをコンピュータ可読プログラム命令によって実装できることが理解されるであろう。

【0063】

これらのコンピュータ可読プログラム命令は、コンピュータまたは他のプログラマブル・データ処理装置のプロセッサを介して実行するそれらの命令がフローチャートもしくはブロック図またはその両方の1つあるいは複数のブロックにおいて指定された機能/動作を実装するための手段を作り出すような、マシンを作り出すために、汎用コンピュータ、専用コンピュータ、または他のプログラマブル・データ処理装置のプロセッサのプロセッサへ提供されてよい。これらのコンピュータ可読プログラム命令は、コンピュータ可読ストレージ媒体に記憶されてもよく、その中に記憶された命令を有するコンピュータ可読ストレージ媒体がフローチャートもしくはブロック図またはその両方の1つまたは複数のブロックにおいて指定された機能/動作の態様を実装する命令を含んだ製造品を備えるような、特定の仕方で機能するようにコンピュータ、プログラマブル・データ装置、もしくは他のデバイスまたはそれらのすべてに命令できる。

【0064】

コンピュータ可読プログラム命令は、コンピュータ、他のプログラマブル装置、または他のデバイス上で実行する命令がフローチャートもしくはブロック図またはその両方の1つまたは複数のブロックにおいて指定された機能/動作を実装するような、コンピュータ実装プロセスを作り出すための一連のオペレーション・ステップがコンピュータ、他のプログラマブル装置、または他のデバイス上で行われるようにするために、コンピュータ、他のプログラマブル・データ処理装置、または他のデバイス上へロードされてもよい。

【 0 0 6 5 】

図中のフローチャートおよびブロック図は、本発明の様々な実施形態によるシステム、方法、およびコンピュータ・プログラム製品の可能な実装のアーキテクチャ、機能性、およびオペレーションを図示する。この点において、フローチャートまたはブロック図中の各ブロックは、指定されたロジック機能（単数または複数）を実装するための1つ以上の実行可能な命令を備えるモジュール、セグメント、または命令の一部を表してよい。いくつかの代替の実装では、ブロック中に記された機能が図中に記された順序以外で発生してよい。例えば、連続して示される2つのブロックが、実際には、実質的に同時に実行されてもよく、または関与する機能性によっては、ブロックがときには逆の順序で実行されてもよい。ブロック図もしくはフローチャート説明図またはその両方の各ブロック、およびブロック図もしくはフローチャート説明図またはその両方におけるブロックの組み合わせを、指定された機能または動作を行うかあるいは専用ハードウェアとコンピュータ命令との組み合わせを実施する専用ハードウェア・ベースのシステムによって実装できることに気付くであろう。

10

【 0 0 6 6 】

図10は、ある実施形態に従って、ストレージ・コントローラ102もしくはホスト104、106、または他の計算デバイス中に含まれてよいいくつかの要素を示すブロック図を図示する。システム1000は、ある実施形態においては少なくとも1つのプロセッサ1004を含む回路素子1002を備えてよい。システム1000は、メモリ1006（例えば、揮発性メモリ・デバイス）、およびストレージ1008も含んでよい。ストレージ1008は、不揮発性メモリ・デバイス（例えば、EEPROM、ROM、PROM、フラッシュ、ファームウェア、プログラマブル・ロジックなど）、磁気ディスク・ドライブ、光ディスク・ドライブ、テープ・ドライブなどを含んでよい。ストレージ1008は、内部ストレージ・デバイス、接続型ストレージ・デバイスもしくはネットワーク・アクセス可能なストレージ・デバイスまたはそれらのすべてを備えてよい。システム1000は、メモリ1006中へロードされて、プロセッサ1004または回路素子1002によって実行されてよいコード1012を含んだプログラム・ロジック1010を含んでよい。ある実施形態において、コード1012を含んだプログラム・ロジック1010は、ストレージ1008に記憶されてよい。ある他の実施形態において、プログラム・ロジック1010は、回路素子1002中に実装されてよい。システム1000中のコンポーネントのうちの1つ以上は、バスを介して、または他の結合もしくは接続1014を介して通信してよい。それゆえに、図10は、プログラム・ロジック1010を他の要素から分離して示すが、メモリ1006もしくは回路素子1002またはその両方にプログラム・ロジック1010が実装されてもよい。

20

30

【 0 0 6 7 】

ある実施形態は、人によるコンピューティング命令または自動処理統合コンピュータ可読コード（automated processing integrating computer-readable code）をコンピューティング・システム中へデプロイするための方法を対象とし、このコードをコンピューティング・システムと組み合わせて、記載される実施形態のオペレーションを行うことが可能である。

40

【 0 0 6 8 】

「ある1つの実施形態（an embodiment）」、「実施形態（embodiment）」、「複数の実施形態（embodiments）」、「その実施形態（the embodiment）」、「それらの複数の実施形態（the embodiments）」、「いくつかの実施形態（some embodiments）」、および「一実施形態（one embodiment）」の用語は、明示的に別に指定されない限り「本発明（単数または複数）の1つ以上の（しかしすべてではない）実施形態」を意味する。

【 0 0 6 9 】

「含んでいる（including）」、「備えている（comprising）」、「有している（having）」の用語およびそれらの変形は、明示的に別に指定されない

50

限り、「含んでいるが限定されない」を意味する。

【0070】

項目の列挙されたりリストは、明示的に別に指定されない限り、これらの項目のいずれかまたはすべてが相互排他的であることを示唆しない。

【0071】

「a(1つの)」、「an(1つの)」、および「the(前記)」の用語は、明確に別に指示されない限り、「1つ以上の」を意味する。

【0072】

互いに通信しているデバイスは、明示的に別に指示されない限り、互いに連続的に通信している必要はない。加えて、互いに通信しているデバイスは、直接的にまたは1つ以上の仲介物を介して間接的に通信してもよい。

10

【0073】

互いに通信しているいくつかのコンポーネントを用いた実施形態の記載は、すべてのかかるコンポーネントが必要なことを示唆しない。逆に、本発明の広範囲にわたる可能な実施形態を示すために種々の随意的なコンポーネントが記載される。

【0074】

さらに、プロセス・ステップ、方法ステップ、アルゴリズムまたは同様のものがシーケンス順に記載されうるが、かかるプロセス、方法およびアルゴリズムが交互順に作動するように構成されてもよい。言い換えれば、記載されうるステップのいずれかのシーケンスまたは順序は、それらのステップがその順序で行われるという必要条件を必ずしも示さない。本明細書に記載されるプロセスのステップは、実用的ないずれかの順序で行われてよい。さらに、いくつかのステップが同時に行われてもよい。

20

【0075】

単一のデバイスまたは物品が本明細書に記載されるときに、すぐに明らかになるのは、単一のデバイス/物品の代わりに1つより多いデバイス/物品が(それらが協働するか否かにかかわらず)用いられてよいことである。同様に、1つより多いデバイスまたは物品が(それらが協働するか否かにかかわらず)本明細書に記載されるときに、すぐに明らかになるのは、単一のデバイスまたは物品が1つより多いデバイスまたは物品の代わりに用いられてもよく、あるいは示される数のデバイスまたはプログラムの代わりに、異なる数のデバイス/物品が用いられてもよいことである。ある1つのデバイスの機能性もしくは特徴またはその両方がかかる機能性/特徴を有するとして明示的には記載されない1つ以上の他のデバイスによって代わりに具現されてもよい。従って、本発明の他の実施形態は、デバイスそれ自体を含む必要がない。

30

【0076】

図に示された少なくとももあるオペレーションは、ある一定の順序で発生するある事象を示す。代替の実施形態においては、あるオペレーションが異なる順序で行われてもよく、修正され、または削除されてもよい。そのうえ、上記のロジックにステップが追加されてもよく、記載される実施形態に依然として適合する。さらに、本明細書に記載されるオペレーションは、順次に発生してもよく、またはあるオペレーションが並列に発生してもよい。なお、さらに、オペレーションは、単一の処理ユニットによって、または分散された処理ユニットによって行われてよい。

40

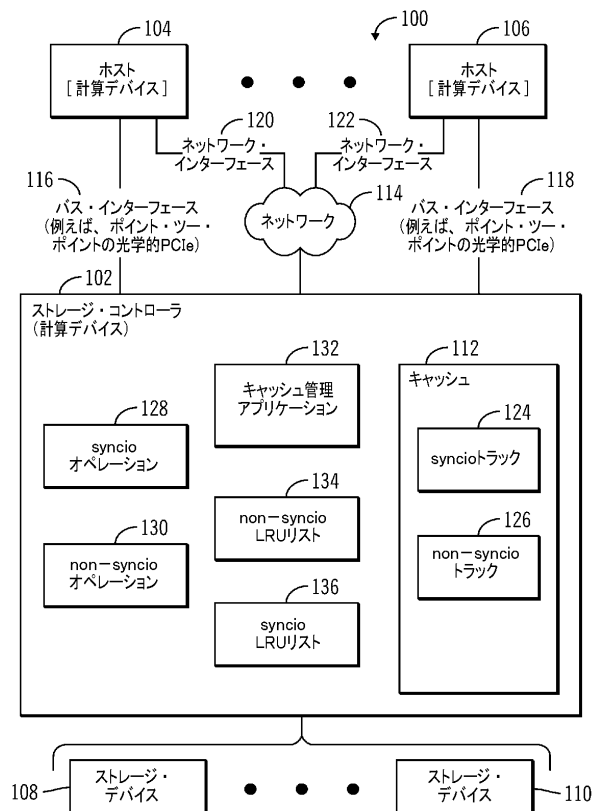
【0077】

本発明の様々な実施形態の先の記載は、例示および説明の目的で提示された。この記載が網羅的であること、または本発明を開示される通りの形態に限定することは意図されない。上記の教示を踏まえて、多くの修正および変更が可能である。意図されるのは、本発明の範囲が、この詳細な記載によってではなく、むしろそれに添付された特許請求の範囲によって限定されることである。上記の明細書、例およびデータは、本発明の構成物の製造および使用の完全な記載を提供する。本発明の範囲から逸脱することなく、本発明の多くの実施形態を作ることができるため。

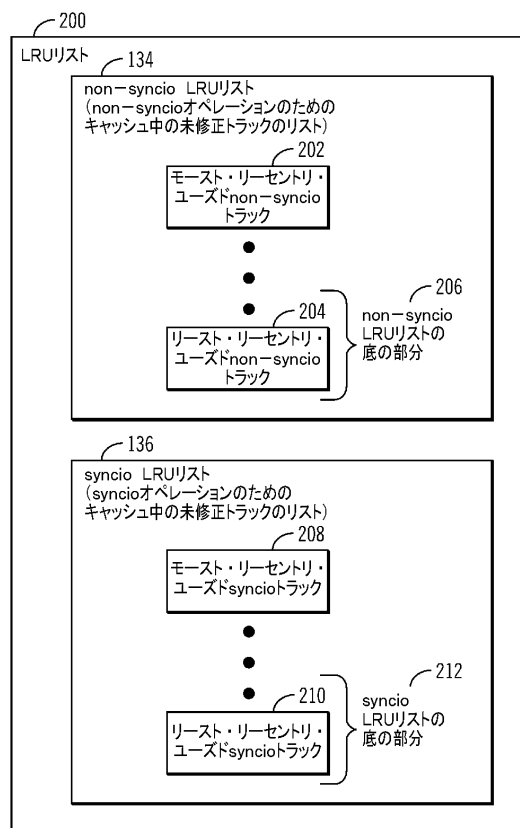
50

【図面】

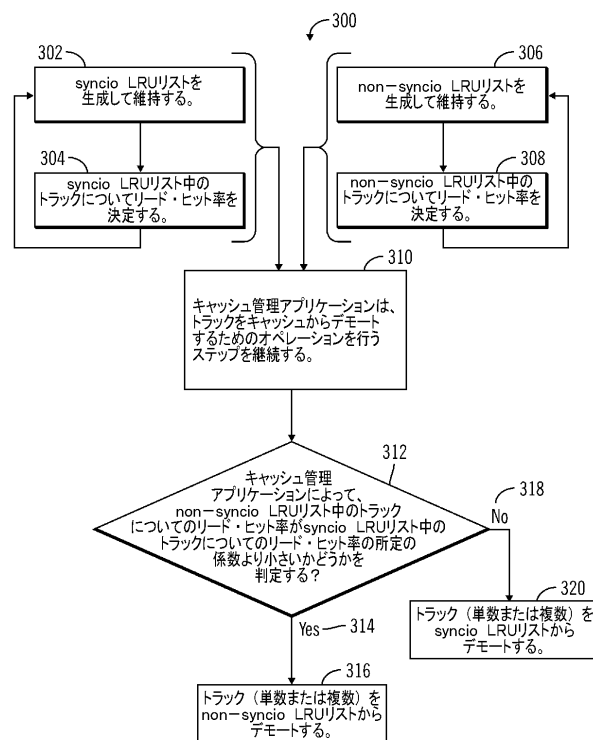
【図 1】



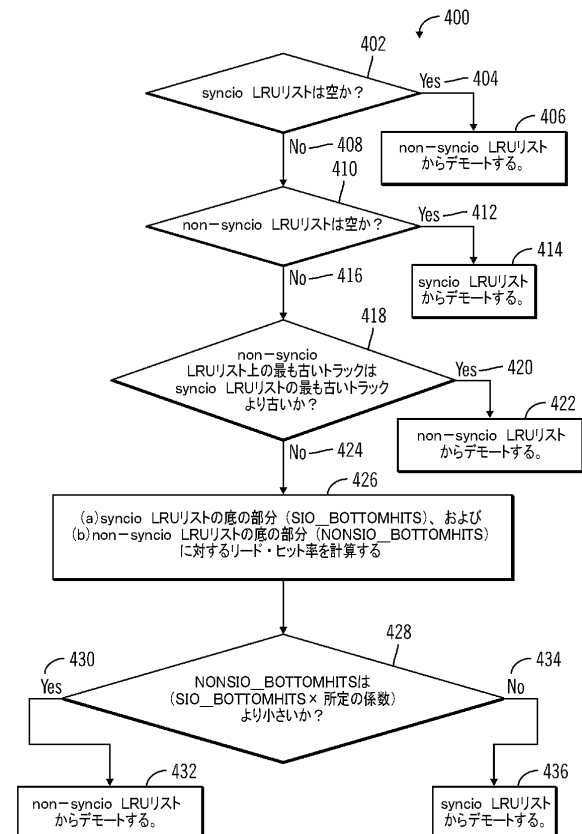
【図 2】



【図 3】



【図 4】



10

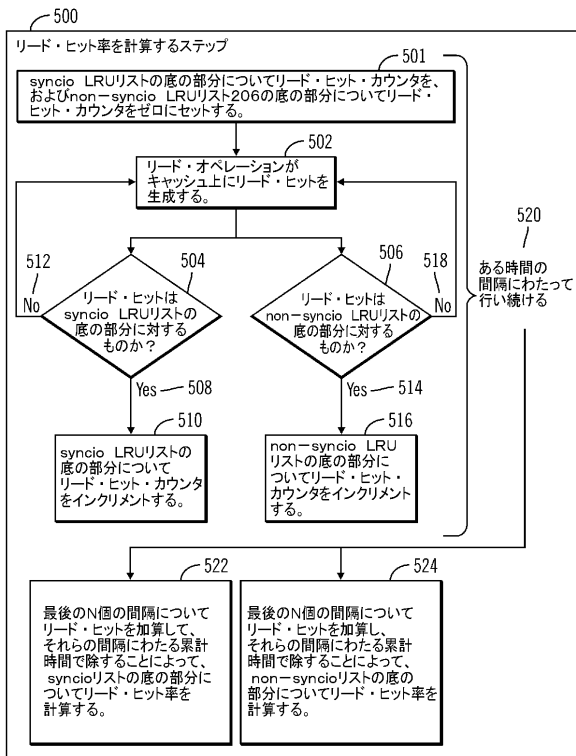
20

30

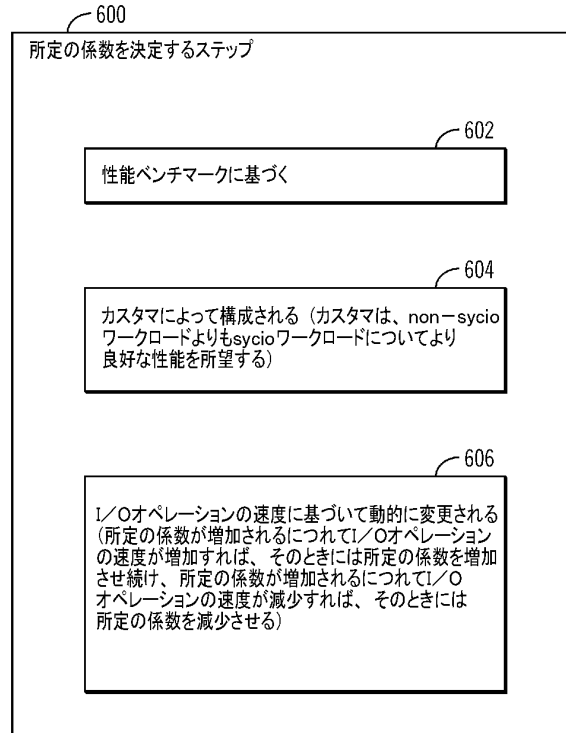
40

50

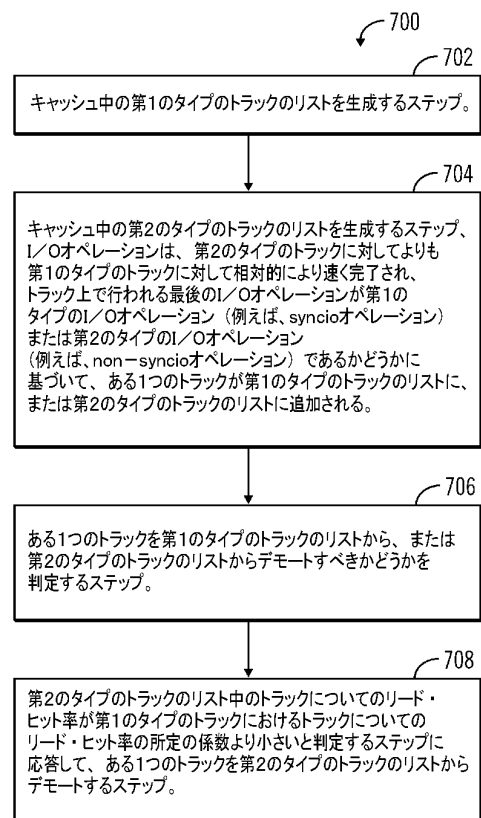
【図 5】



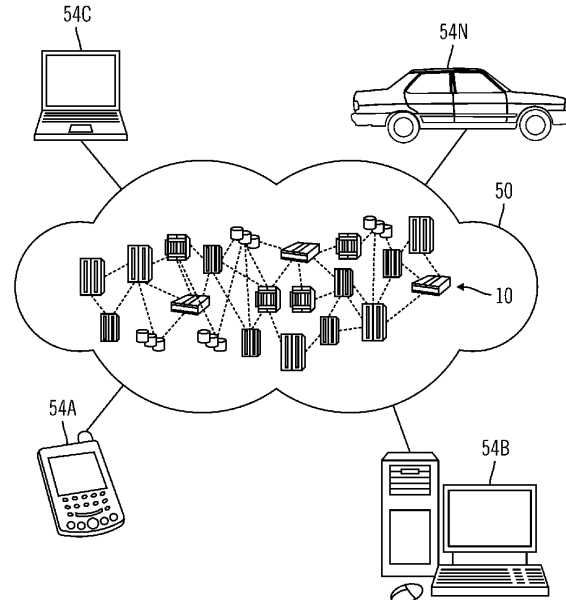
【図 6】



【図 7】



【図 8】



10

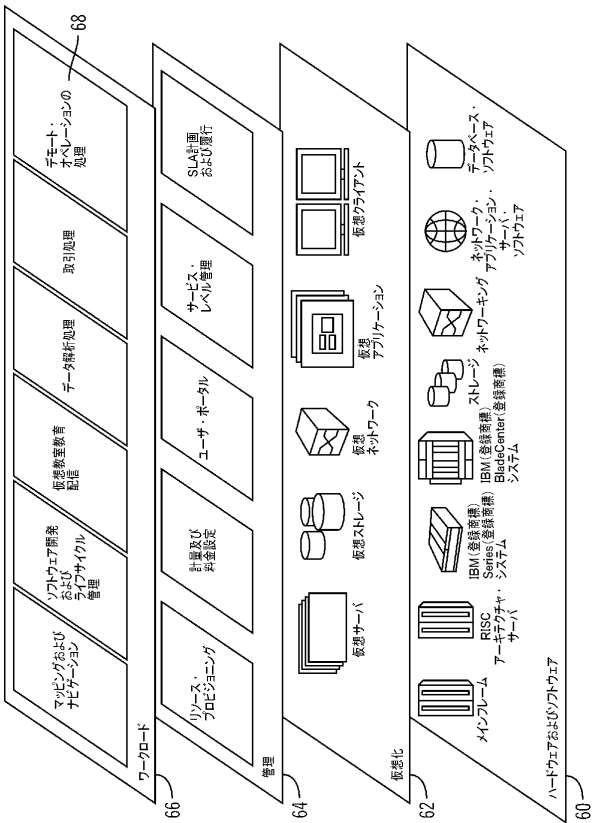
20

30

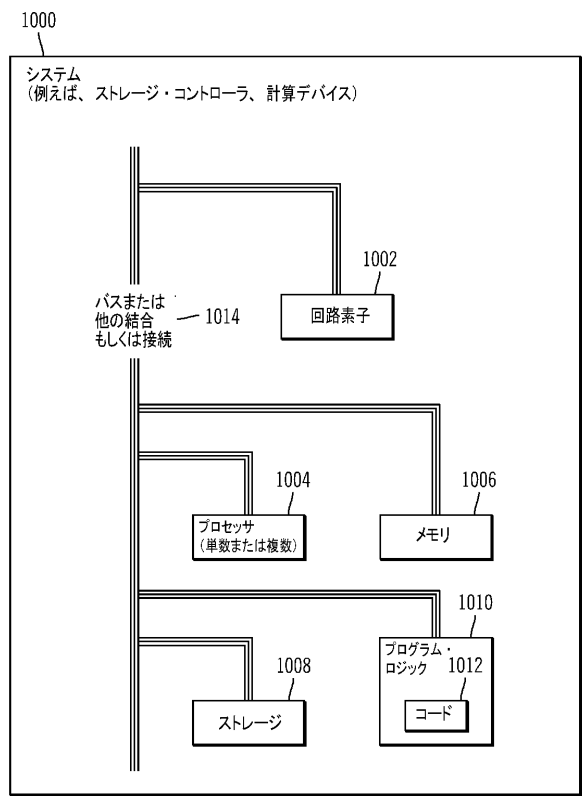
40

50

【図 9】



【図 10】



10

20

30

40

50

フロントページの続き

- (72)発明者 グプタ、ロケシュ
 アメリカ合衆国 8 5 7 4 4 - 0 0 0 2 アリゾナ州 ツーソン サウス・リタ・ロード 9 0 0 0
- (72)発明者 アッシュ、ケビン、ジョン
 アメリカ合衆国 8 5 7 4 4 - 0 0 0 2 アリゾナ州 ツーソン サウス・リタ・ロード 9 0 0 0
- (72)発明者 アンダーソン、カイル
 アメリカ合衆国 8 5 7 4 4 - 0 0 0 2 アリゾナ州 ツーソン サウス・リタ・ロード 9 0 0 0
- 審査官 後藤 彰
- (56)参考文献 米国特許出願公開第 2 0 1 7 / 0 0 6 8 6 2 4 (U S , A 1)
 米国特許出願公開第 2 0 1 7 / 0 1 3 9 8 4 1 (U S , A 1)
- (58)調査した分野 (Int.Cl., D B 名)
- G 0 6 F 1 2 / 1 2 3
 G 0 6 F 1 2 / 0 6
 G 0 6 F 1 2 / 0 8 6 6
 G 0 6 F 3 / 0 6