

(19) 日本国特許庁(JP)

(12) 公開特許公報(A)

(11) 特許出願公開番号

特開2005-339198  
(P2005-339198A)

(43) 公開日 平成17年12月8日(2005.12.8)

(51) Int. Cl. <sup>7</sup>	F I	テーマコード (参考)
G06F 3/06	G06F 3/06 304N	5B005
G06F 12/08	G06F 3/06 302A	5B065
G06F 12/12	G06F 12/08 543B	
	G06F 12/12 551	

審査請求 有 請求項の数 18 O L (全 33 頁)

(21) 出願番号	特願2004-157103 (P2004-157103)	(71) 出願人	390009531 インターナショナル・ビジネス・マシー ズ・コーポレーション INTERNATIONAL BUSIN ESS MASCHINES 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
(22) 出願日	平成16年5月27日 (2004.5.27)	(74) 代理人	100086243 弁理士 坂口 博
(特許庁注：以下のものは登録商標)		(74) 代理人	100091568 弁理士 市位 嘉宏
1. JAVA		(74) 代理人	100108501 弁理士 上野 剛史

最終頁に続く

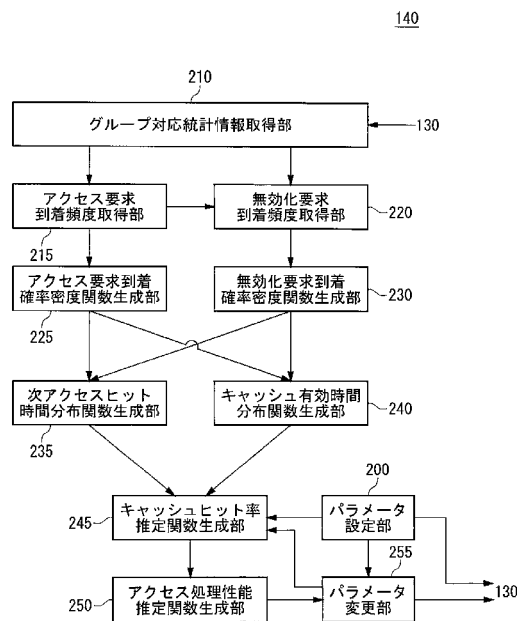
(54) 【発明の名称】 キャッシュヒット率推定装置、キャッシュヒット率推定方法、プログラム及び記録媒体

(57) 【要約】

【課題】 キャッシュ装置のキャッシュヒット率を解析的に精度良く求める。

【解決手段】 要求元装置からアクセスされたアクセス対象データをキャッシュするキャッシュ装置のキャッシュヒット率を推定するキャッシュヒット率推定装置であって、それぞれのアクセス対象データに対するアクセス要求について計測された平均到着頻度を取得するアクセス要求到着頻度取得部と、それぞれのアクセス対象データに対するアクセス要求の平均到着頻度に基づいて、当該アクセス対象データに対するアクセス要求の到着時間間隔の確率密度関数であるアクセス要求到着確率密度関数を生成するアクセス要求到着確率密度関数生成部と、複数のアクセス対象データのアクセス要求到着確率密度関数に基づいて、それぞれのアクセス対象データのキャッシュヒット率の推定関数を生成するキャッシュヒット率推定関数生成部とを備えるキャッシュヒット率推定装置を提供する。

【選択図】 図2



## 【特許請求の範囲】

## 【請求項 1】

要求元装置からアクセスされたアクセス対象データをキャッシュするキャッシュ装置のキャッシュヒット率を推定するキャッシュヒット率推定装置であって、

複数の前記アクセス対象データの少なくとも1つは、キャッシュミス時のアクセスコストが他の前記アクセス対象データと異なるものであり、

それぞれの前記アクセス対象データに対するアクセス要求について計測された平均到着頻度を取得するアクセス要求到着頻度取得部と、

それぞれのアクセス対象データに対するアクセス要求の平均到着頻度に基づいて、当該アクセス対象データに対するアクセス要求の到着時間間隔の確率密度関数であるアクセス要求到着確率密度関数を生成するアクセス要求到着確率密度関数生成部と、

前記複数のアクセス対象データの前記アクセス要求到着確率密度関数に基づいて、それぞれの前記アクセス対象データのキャッシュヒット率の推定関数を生成するキャッシュヒット率推定関数生成部と

を備えるキャッシュヒット率推定装置。

10

## 【請求項 2】

前記要求元装置からのアクセス要求に対するアクセス処理性能を変化させる設定パラメータを前記キャッシュ装置に設定するパラメータ設定部と、

それぞれの前記アクセス対象データの前記キャッシュヒット率推定関数及び前記キャッシュミス時のアクセスコストに基づいて、前記アクセス処理性能の推定関数を生成するアクセス処理性能推定関数生成部と、

前記キャッシュ装置に設定された前記設定パラメータを、前記アクセス処理性能推定関数による前記アクセス処理性能の推定値をより高める前記設定パラメータに変更するパラメータ変更部と

を更に備える請求項 1 記載のキャッシュヒット率推定装置。

20

## 【請求項 3】

前記キャッシュヒット率推定関数生成部及び前記アクセス処理性能推定関数生成部は、それぞれの前記アクセス対象データをキャッシュするか否かを設定する設定パラメータを入力とする前記キャッシュヒット率推定関数及び前記アクセス処理性能推定関数を生成し、

前記パラメータ変更部は、一の前記アクセス対象データをキャッシュする場合に、キャッシュしない場合と比較して前記アクセス処理性能の推定値がより高くなることを条件として、前記一の前記アクセス対象データをキャッシュすることを指定する前記設定パラメータを前記キャッシュ装置に設定する

請求項 2 記載のキャッシュヒット率推定装置。

30

## 【請求項 4】

前記キャッシュヒット率推定関数生成部及び前記アクセス処理性能推定関数生成部は、それぞれの前記アクセス対象データをキャッシュする優先度を入力とする前記キャッシュヒット率推定関数及び前記アクセス処理性能推定関数を生成し、

前記パラメータ変更部は、一の前記アクセス対象データをキャッシュする優先度を第 1 の優先度とした場合に、第 2 の優先度とした場合と比較して前記アクセス処理性能の推定値がより高くなることを条件として、前記一の前記アクセス対象データをキャッシュする優先度を前記第 1 の優先度とする前記設定パラメータを前記キャッシュ装置に設定する

請求項 2 記載のキャッシュヒット率推定装置。

40

## 【請求項 5】

前記キャッシュ装置は、前記複数のアクセス対象データのそれぞれを、当該キャッシュ装置のキャッシュ記憶領域を分割した複数のキャッシュ分割領域のうち、当該アクセス対象データに対応する予め定められたキャッシュ分割領域にキャッシュし、

前記キャッシュヒット率推定関数生成部及び前記アクセス処理性能推定関数生成部は、前記キャッシュ記憶領域に対する前記複数のキャッシュ分割領域の割合の組み合わせを入

50

力とする前記キャッシュヒット率推定関数及び前記アクセス処理性能推定関数を生成し、  
前記パラメータ変更部は、前記複数のキャッシュ分割領域の割合の組み合わせを第1の  
組み合わせとした場合に、第2の組み合わせとした場合と比較して前記アクセス処理性能  
の推定値がより高くなることを条件として、前記複数のキャッシュ分割領域の割合の組み  
合わせを第1の組み合わせとする前記設定パラメータを前記キャッシュ装置に設定する  
請求項2記載のキャッシュヒット率推定装置。

【請求項6】

前記キャッシュ装置は、それぞれの前記アクセス対象データを、当該キャッシュ装置の  
キャッシュ記憶領域を分割した複数のキャッシュ分割領域のうち、当該アクセス対象デー  
タに対応する予め定められたキャッシュ分割領域にキャッシュし、

10

前記キャッシュヒット率推定関数生成部及び前記アクセス処理性能推定関数生成部は、  
それぞれの前記アクセス対象データをいずれの前記キャッシュ分割領域にキャッシュする  
かの対応を指定する組み合わせを入力とする前記キャッシュヒット率推定関数及び前記ア  
クセス処理性能推定関数を生成し、

前記パラメータ変更部は、前記対応を第1の組み合わせとした場合に、第2の組み合わ  
せとした場合と比較して前記アクセス処理性能の推定値がより高くなることを条件として  
、前記対応を第1の組み合わせとする前記設定パラメータを前記キャッシュ装置に設定す  
る

請求項2記載のキャッシュヒット率推定装置。

【請求項7】

20

それぞれの前記アクセス対象データについて、前記アクセス要求到着確率密度関数に基  
づいて、当該アクセス対象データへの前記アクセス要求を受信した後、当該アクセス対象  
データがキャッシュから無効化される前に当該アクセス対象データに対する次の前記アク  
セス要求を受信する確率の時間分布である次アクセスヒット時間分布関数を生成する次ア  
クセスヒット時間分布関数生成部と、

それぞれの前記アクセス対象データについて、前記アクセス要求到着確率密度関数に基  
づいて、予め定められた時点を基準とし、当該アクセス対象データに対する少なくとも1  
つの前記アクセス要求を受信し、かつ、当該アクセス対象データがキャッシュから無効化  
されていない確率の時間分布であるキャッシュ有効時間分布関数を生成するキャッシュ有  
効時間分布関数生成部を更に備え、

30

前記キャッシュヒット率推定関数生成部は、それぞれの前記アクセス対象データについ  
て、前記複数のアクセス対象データの前記キャッシュ有効時間分布関数と、当該アクセス  
対象データに対する前記次アクセスヒット時間分布関数とに基づいて、当該アクセス対象  
データの前記キャッシュヒット率推定関数を生成する

請求項2記載のキャッシュヒット率推定装置。

【請求項8】

それぞれの前記アクセス対象データについて、当該アクセス対象データに対する無効化  
要求の平均到着頻度に基づいて、無効化要求の到着時間間隔の確率密度関数である無効化  
要求到着確率密度関数を生成する無効化要求到着確率密度関数生成部を更に備え、

前記次アクセスヒット時間分布関数生成部及び前記キャッシュ有効時間分布関数生成部  
は、前記無効化要求到着確率密度関数に更に基づいて、前記次アクセスヒット時間分布関  
数及び前記キャッシュ有効時間分布関数を生成する

40

請求項7記載のキャッシュヒット率推定装置。

【請求項9】

前記キャッシュ装置は、キャッシュしている前記アクセス対象データに対して無効化要  
求を受信した場合、又は、当該アクセス対象データをキャッシュしてから当該アクセス対  
象データに対して前記アクセス要求を受けることなく予め定められた時間が経過した場合  
に当該アクセス対象データをキャッシュから無効化し、

前記次アクセスヒット時間分布関数生成部及び前記キャッシュ有効時間分布関数生成部  
は、前記予め定められた時間の経過前及び経過後の各区間について前記次アクセスヒット

50

時間分布関数及び前記キャッシュ有効時間分布関数を生成する

請求項 8 記載のキャッシュヒット率推定装置。

【請求項 10】

前記キャッシュ装置は、最大 K 個の前記アクセス対象データを L R U (Least Recently Used) 方式によりキャッシュし、

前記キャッシュヒット率推定関数生成部は、

前記複数のアクセス対象データの前記キャッシュ有効時間分布関数に基づいて、無効化されることなくキャッシュされている前記アクセス対象データの数の期待値を求めるエントリ数期待値関数を生成するエントリ数期待値関数生成部と、

前記エントリ数期待値関数に基づいて、予め定められた時点を基準として、K 個の前記アクセス対象データがキャッシュされるまでのキャッシュ充填時間の期待値を算出するキャッシュ充填時間算出部と、

それぞれの前記アクセス対象データについて、当該アクセス対象データに対するアクセス要求を受信してから前記キャッシュ充填時間の期待値により定められる時間の経過までに当該アクセス対象データに対する次のアクセス要求を受信する確率の関数を前記アクセスヒット時間分布関数に基づいて算出し、当該アクセス対象データの前記キャッシュヒット率推定関数とする次アクセスヒット率推定関数生成部と

を有する

請求項 7 記載のキャッシュヒット率推定装置。

【請求項 11】

それぞれの前記アクセス対象データには、当該アクセス対象データをキャッシュする優先度が定められており、

前記キャッシュ装置は、

キャッシュされた前記アクセス対象データを L R U 方式により管理する、優先度毎に設けられた複数の L R U リストと、

アクセス要求を受けた前記アクセス対象データを、当該アクセス対象データの優先度に対応する前記 L R U リストの先頭に移動するエントリ移動部と、

優先度が最も低い前記 L R U リストが空である場合に、前記複数の L R U リストのそれぞれに登録された前記アクセス対象データを、より優先度が低い前記 L R U リストに登録し直すシフト処理を行う L R U リストシフト部と

を備え、

前記キャッシュヒット率推定関数生成部は、

前記複数のアクセス対象データの前記キャッシュ有効時間分布関数に基づいて、前記複数の L R U リストに登録された前記アクセス対象データの合計数の期待値を求めるエントリ数期待値関数を生成するエントリ数期待値関数生成部と、

前記エントリ数期待値関数に基づいて、前記シフト処理の周期の期待値を算出するシフト周期算出部と、

前記シフト処理の周期を複数に等分した各時点において前記アクセス対象データへのアクセス要求を受けてから当該アクセス対象データが他の前記アクセス対象データに置換されるまでのリプレース時間の期待値を算出するリプレース時間算出部と、

それぞれの前記アクセス対象データについて、前記シフト処理の周期を複数に等分した各時点において当該アクセス対象データに対するアクセス要求を受信してから前記リプレース時間の期待値により定められる時間の経過までに当該アクセス対象データに対する次のアクセス要求を受信する確率を算出する次アクセスヒット率推定関数を、前記アクセスヒット時間分布関数及び前記リプレース時間に基づいて生成する次アクセスヒット率推定関数生成部と、

それぞれの前記アクセス対象データについて、前記シフト処理の周期を複数に等分した前記複数の時点のそれぞれにおいて当該アクセス対象データに対するアクセス要求を受信した場合の前記次アクセスヒット率推定関数を全ての前記時点について平均し、当該アクセス対象データの前記キャッシュヒット率推定関数とする次アクセスヒット率平均化部と

を有する

請求項 7 記載のキャッシュヒット率推定装置。

【請求項 1 2】

前記アクセス対象データは、前記アクセス要求を受信したサーバ装置上で当該アクセス要求に対応するサーバプログラムを、前記アクセスコストにより指定される処理時間実行することにより生成され、

前記キャッシュ装置は、前記サーバ装置により生成された前記アクセス対象データをキャッシュするキャッシュ記憶領域を備える

請求項 2 記載のキャッシュヒット率推定装置。

【請求項 1 3】

それぞれの前記アクセス対象データは、複数のグループのいずれかに属し、

それぞれの前記グループについて、当該グループに属する全ての前記アクセス対象データについて計測された平均キャッシュヒット率を取得するグループ対応キャッシュヒット率取得部と、

それぞれの前記グループについて、当該グループに属する全ての前記アクセス対象データのうちキャッシュしている前記アクセス対象データの個数であるグループ対応エントリ数の計測値を取得するグループ対応エントリ数取得部と、

それぞれの前記グループについて、当該グループの前記グループ対応エントリ数を当該グループの前記平均キャッシュヒット率で割ることにより、当該グループに属する前記アクセス対象データの数の推定値を算出するアクセス対象データ数算出部と

を備え、

前記アクセス要求到着頻度取得部は、それぞれの前記グループについて、当該グループに属する全アクセス対象データについて計測された前記アクセス要求の平均到着頻度を当該グループに属する前記アクセス対象データの数の推定値で割ることにより、当該アクセス対象データの平均到着頻度を取得し、

前記キャッシュヒット率推定関数生成部は、それぞれの前記グループに属する前記アクセス対象データの前記キャッシュヒット率推定関数を、当該グループに属する前記アクセス対象データの数の推定値に更に基づいて生成する

請求項 2 記載のキャッシュヒット率推定装置。

【請求項 1 4】

複数の前記キャッシュ装置におけるそれぞれの前記グループを、一の前記キャッシュ装置における対応する前記グループに統合した場合における前記アクセス処理性能推定関数を生成する統合時間関数生成部と、

前記統合時間関数生成部により生成された前記アクセス処理性能推定関数による前記アクセス処理性能の推定値が、前記複数のキャッシュ装置の前記アクセス処理性能の推定値の合計より高い場合に、前記複数のキャッシュ装置におけるそれぞれの前記グループを、前記一のキャッシュ装置における対応する前記グループに統合するグループ統合部と

を更に備える

請求項 1 3 記載のキャッシュヒット率推定装置。

【請求項 1 5】

一の前記キャッシュ装置におけるそれぞれの前記グループを、複数の前記キャッシュ装置における対応する前記グループに分割した場合における前記複数のキャッシュ装置のそれぞれの前記アクセス処理性能推定関数を生成する分割時間関数生成部と、

前記分割時間関数生成部により生成された前記アクセス処理性能推定関数による前記アクセス処理性能の推定値の合計が、前記一のキャッシュ装置の前記アクセス処理性能の推定値より高い場合に、前記一の前記キャッシュ装置におけるそれぞれの前記グループを、複数の前記キャッシュ装置における対応する前記グループに分割するグループ分割部と

を更に備える

請求項 1 3 記載のキャッシュヒット率推定装置。

【請求項 1 6】

10

20

30

40

50

要求元装置からアクセスされたアクセス対象データをキャッシュするキャッシュ装置のキャッシュヒット率をコンピュータにより推定するキャッシュヒット率推定方法であって

、  
複数の前記アクセス対象データの少なくとも1つは、キャッシュミス時のアクセスコストが他の前記アクセス対象データと異なるものであり、

それぞれの前記アクセス対象データに対するアクセス要求について計測された平均到着頻度を取得するアクセス要求到着頻度取得段階と、

それぞれのアクセス対象データに対するアクセス要求の平均到着頻度に基づいて、当該アクセス対象データに対するアクセス要求の到着時間間隔の確率密度関数であるアクセス要求到着確率密度関数を生成するアクセス要求到着確率密度関数生成段階と、

前記複数のアクセス対象データの前記アクセス要求到着確率密度関数に基づいて、それぞれの前記アクセス対象データのキャッシュヒット率の推定関数を生成するキャッシュヒット率推定関数生成段階と

を備えるキャッシュヒット率推定方法。

#### 【請求項17】

要求元装置からアクセスされたアクセス対象データをキャッシュするキャッシュ装置のキャッシュヒット率を推定するキャッシュヒット率推定装置用のプログラムであって、

複数の前記アクセス対象データの少なくとも1つは、キャッシュミス時のアクセスコストが他の前記アクセス対象データと異なるものであり、

当該プログラムは、前記キャッシュヒット率推定装置を、

それぞれの前記アクセス対象データに対するアクセス要求について計測された平均到着頻度を取得するアクセス要求到着頻度取得部と、

それぞれのアクセス対象データに対するアクセス要求の平均到着頻度に基づいて、当該アクセス対象データに対するアクセス要求の到着時間間隔の確率密度関数であるアクセス要求到着確率密度関数を生成するアクセス要求到着確率密度関数生成部と、

前記複数のアクセス対象データの前記アクセス要求到着確率密度関数に基づいて、それぞれの前記アクセス対象データのキャッシュヒット率の推定関数を生成するキャッシュヒット率推定関数生成部と

して機能させるプログラム。

#### 【請求項18】

請求項17に記載のプログラムを記録した、コンピュータにより読み取り可能な記録媒体。

#### 【発明の詳細な説明】

#### 【技術分野】

#### 【0001】

本発明は、キャッシュヒット率推定装置、キャッシュヒット率推定方法、プログラム及び記録媒体に関する。特に本発明は、キャッシュミス時のアクセスコストが異なる複数のアクセス対象データのそれぞれについてのキャッシュヒット率を推定するキャッシュヒット率推定装置、キャッシュヒット率推定方法、プログラム及び記録媒体に関する。

#### 【背景技術】

#### 【0002】

近年、Webアプリケーション等を実行するサーバ装置は、例えばHTTPサーバ、Webコンテナ、EJBコンテナ、及びデータベースサーバ等の各コンポーネント毎にアクセスされたデータをキャッシュし、負荷の軽減及び応答時間の低減を図っている。このようなキャッシュを用いるアプリケーションの処理性能は、アクセス対象データのキャッシュヒット率、キャッシュヒット時の応答時間、及び、キャッシュミス時のアクセス処理時間により定まる。しかしながら、従来においては、キャッシュの各設定パラメータがキャッシュヒット率に与える影響が定量的に明らかでなく、これらのパラメータをヒューリスティックに設定していた。

#### 【0003】

10

20

30

40

50

また従来、LRU (Least Recently Used) 方式によるキャッシュのヒット率を推定する方法として、非特許文献 1 から 4 が提案されている。

非特許文献 1 は、アクセス要求を受けた際に、キャッシュする対象となる各アイテムが LRU リストの  $n$  番目に存在する確率を、以下の式 (1) により求める。ここで、 $C^N$  は全てのアイテムについての参照コスト (アクセス要求を受けた際のアクセス対象アイテムのリスト内における位置)、 $q_r$  はアイテム  $r$  ( $r=1, 2, \dots, N$ ) がアクセスされる確率である。

【数 1】

$$P[C^N = n] = \sum_{r=1}^N \sum_{a=0}^{n-1} (-1)^{n-1-a} \binom{N-1-a}{n-1-a} \sum_{A: |A|=a, r \notin A} \frac{q_r^2}{(1-Q_A)} \quad \text{where } Q_A \equiv \sum_{r \in A} q_r \quad (1) \quad 10$$

【0004】

式 (1) を用いれば、 $k$  個のアイテムをキャッシュする LRU キャッシュのヒット率は、 $P[CN \leq k]$  により求めることができる。

【0005】

非特許文献 2 及び非特許文献 3 は、アイテム毎の参照コスト  $C_r^N$  の確率母関数を以下の式 (2) により求める。

【数 2】

$$E[z^{C^N}] = \sum_{i=1}^N q_i E[z^{C_i^N}] \quad 20$$

$$E[z^{C_r^N}] = \int_0^\infty q_r e^{-q_r t} \left[ \prod_{j \in [N], j \neq r} (e^{-q_j t} + (1 - e^{-q_j t}) z) \right] dt \quad (2)$$

【0006】

非特許文献 4 は、アイテム数  $N$  及びキャッシュサイズ  $k$  が無限大に近づく時の漸近的な近似式として以下の式 (3-1) 及び (3-2) を示す。式 (3-1) はアイテムの参照確率分布が多項式で減少していく場合 (Heavy Tail) の近似式、式 (3-2) は参照確率分布が指数的に減少していく場合 (Light Tail) の近似式である。さらに  $N, k$  が比較的小さな値であっても、この近似式の精度が高いことを実験により検証している。

【数 3】

$$\lim_{k \rightarrow \infty} \frac{P[C^\infty > k]}{P[R^\infty > k]} = \left(1 - \frac{1}{\alpha}\right) \left[\Gamma\left(1 - \frac{1}{\alpha}\right)\right] \quad \text{if } P[R^\infty = n] \approx c/n^\alpha \quad (3-1) \quad 30$$

$$\lim_{k \rightarrow \infty} \frac{P[C^\infty > k]}{P[R^\infty > k]} = e^\gamma \quad \text{if } P[R^\infty = n] \approx ce^{-\lambda n \beta} \quad (3-2) \quad 40$$

【非特許文献 1】P. J. Burville and J. F. C. Kingman, "On a model for strage and search.", Journal of Applied Probability, 10: p.697-701, 1973 年

【非特許文献 2】P. Flajolet, D. Gardy, and L. Thimonier, "Birthday paradox, coup on collector, caching algorithms and self-organizing search.", Discrete Applied mathematics, 39: p.207-229, 1992 年

【非特許文献 3】J. A. Fill and L. Holst, "On the distribution search cost for the move-to-front rule.", Random Structures and algorithms, 8(3): p.179-186, 1996 年

【非特許文献 4】P. R. Jelenkovic, "Asymptotic approximation of the move-to-front 50

search cost distribution and least-recently-used caching fault probabilities.",  
Annals of Applied Probability, 9(2): p.430-464, 1999年.

【発明の開示】

【発明が解決しようとする課題】

【0007】

キャッシュの設定パラメータをヒューリスティックに設定する場合、アプリケーションの実行環境に応じた最適値を求めるのが困難である。すなわち例えば、設定パラメータの最適値を求めるためには、情報システムの構築時等において、設定パラメータを変えながら性能測定を繰り返して最適な設定パラメータを求める方法が考えられるが、作業の負担が大きく動的に変化するアプリケーション実行環境に対応することができない。

10

【0008】

そこで本発明は、キャッシュの性能をより高めるため、アプリケーション実行中に計測した各種の統計情報に基づいてより適切な設定パラメータを算出し、キャッシュに設定するキャッシュシステムを提案する。より具体的には、各種の統計情報に基づいてアクセス処理の性能を推定する推定関数を生成し、アクセス処理性能の推定値を高める設定パラメータを求めてキャッシュに設定する。

【0009】

ここで、Webアプリケーション等を実行するサーバ装置においては、キャッシュする対象となるデータは、各種のサーバアプリケーションが出力するデータであり、アクセス対象データによってキャッシュミス時のアクセスコストが異なる。したがって、アクセス

20

処理性能を精度良く推定するためには、アクセス対象データ毎のキャッシュヒット率を求める必要がある。

【0010】

非特許文献1においては、アイテムの数が増加すると、式(1)におけるアイテムの組み合わせAの数が爆発的に増加するため、アクセス処理性能の評価を効率良く行うことが困難である。

【0011】

また非特許文献2及び3においても、アイテムの数が増加すると数值的に計算困難となるため、式(2)を用いてアクセス処理性能の評価を行うことは困難である。

【0012】

また非特許文献4においては、アイテム毎のヒット率については考慮されていない。しかし、本発明の対象となるアプリケーション実行環境においては、アクセス対象データ毎のキャッシュヒット率が要求されるため、式(3)を用いてアクセス処理性能の評価を行うことは困難である。

30

【0013】

そこで本発明は、上記の課題を解決することのできるキャッシュヒット率推定装置、キャッシュヒット率推定方法、プログラム及び記録媒体を提供することを目的とする。この目的は特許請求の範囲における独立項に記載の特徴の組み合わせにより達成される。また従属項は本発明の更なる有利な具体例を規定する。

【課題を解決するための手段】

40

【0014】

本発明の第1の形態によると、要求元装置からアクセスされたアクセス対象データをキャッシュするキャッシュ装置のキャッシュヒット率を推定するキャッシュヒット率推定装置であって、複数の前記アクセス対象データの少なくとも1つは、キャッシュミス時のアクセスコストが他の前記アクセス対象データと異なるものであり、それぞれの前記アクセス対象データに対するアクセス要求について計測された平均到着頻度を取得するアクセス要求到着頻度取得部と、それぞれのアクセス対象データに対するアクセス要求の平均到着頻度に基づいて、当該アクセス対象データに対するアクセス要求の到着時間間隔の確率密度関数であるアクセス要求到着確率密度関数を生成するアクセス要求到着確率密度関数生成部と、前記複数のアクセス対象データの前記アクセス要求到着確率密度関数に基づいて

50



、それぞれの前記アクセス対象データのキャッシュヒット率の推定関数を生成するキャッシュヒット率推定関数生成部とを備えるキャッシュヒット率推定装置と、当該キャッシュヒット率推定装置に関するキャッシュヒット率推定方法、プログラム、及び記録媒体とを提供する。

【0015】

なお、上記の発明の概要は、本発明の必要な特徴の全てを列挙したものではなく、これらの特徴群のサブコンビネーションもまた、発明となりうる。

【発明の効果】

【0016】

本発明によれば、キャッシュ装置のキャッシュヒット率を解析的に精度良く求めることができる。そして、解析結果に基づいてキャッシュ装置の設定パラメータを最適化することにより、アクセス処理性能を高めることができる。

【発明を実施するための最良の形態】

【0017】

以下、発明の実施の形態を通じて本発明を説明するが、以下の実施形態は特許請求の範囲にかかる発明を限定するものではなく、また実施形態の中で説明されている特徴の組み合わせの全てが発明の解決手段に必須であるとは限らない。

【0018】

図1は、本実施形態に係る情報システム10の構成を示す。情報システム10は、1又は複数の要求元装置100と、1又は複数のサーバ装置110と、キャッシュシステム120とを備える。

要求元装置100は、アクセス要求をサーバ装置110に対して送信することにより、アクセスする対象となるアクセス対象データの送信を要求する。そして、要求元装置100は、サーバ装置110からアクセス対象データを受信する。本実施形態に係る要求元装置100は、例えばクライアント端末であり、HTTPやSOAP等に基づくアクセス要求をサーバ装置110に対して送信する。これに代えて要求元装置100は、例えばキャッシュメモリを介してメモリをアクセスするCPU等であってもよい。

【0019】

サーバ装置110は、キャッシュシステム120を介して要求元装置100からのアクセス要求を受信し、アクセス要求に対応する処理を行って、アクセス対象データを生成する。そしてサーバ装置110は、アクセス対象データを、キャッシュシステム120を介して要求元装置100へ返信する。本実施形態に係るサーバ装置110は、例えばWebアプリケーションサーバであり、アクセス要求に対応するサブレットやJSP等のサーバプログラムを実行して、当該アクセス要求に対応するアクセス対象データを生成する。これに代えてサーバ装置110は、例えばCPU等の要求元装置100からのアクセス要求を受けて記憶領域をアクセスするメモリシステムやI/Oシステム等であってもよい。また、複数のサーバ装置110のそれぞれは、複数のentity Beansを分割した一部のentity Beansが割り当てられたEJBコンテナを処理するJVM(Java Virtual Machine)等であってもよい。

【0020】

キャッシュシステム120は、キャッシュ装置130と、キャッシュ最適化装置140とを有する。キャッシュ装置130は、要求元装置100からアクセスされ、サーバ装置110により返信されたアクセス対象データをキャッシュ記憶領域135にキャッシュする。そして、キャッシュしたアクセス対象データをアクセスするアクセス要求を受けた場合に、キャッシュしたアクセス対象データをキャッシュ記憶領域135から要求元装置100へ返信する。本実施形態において、キャッシュ装置130は、最大K個のアクセス対象データをLRU方式によりキャッシュする。

【0021】

キャッシュ最適化装置140は、本発明に係るキャッシュヒット率推定装置の一例であり、情報システム10による処理の実行中に、1又は複数の要求元装置100からサーバ

10

20

30

40

50

装置 110 に対するアクセス要求の到着頻度や、キャッシュされたアクセス対象データの無効化頻度等の計測値を取得する。次に、キャッシュ最適化装置 140 は、これらの計測値に基づいて、キャッシュ装置 130 のキャッシュヒット率を推定し、キャッシュヒット率の推定関数に基づいてキャッシュ装置 130 によるアクセス処理性能の推定関数を生成する。そして、アクセス処理性能の推定値をより高める設定パラメータを算出し、キャッシュ装置 130 に設定する。このようにキャッシュ最適化装置 140 は、実環境に基づいてキャッシュ装置 130 のアクセス処理性能をモデル化し、当該モデルを用いて最適な設定パラメータをキャッシュ装置 130 に設定することにより、実環境に応じてキャッシュ装置 130 のアクセス処理性能を高めることができる。

#### 【0022】

以上において、1又は複数のサーバ装置 110 は、アクセス要求を受信すると、当該アクセス要求に対応するサーバプログラムをある処理時間実行して、アクセス対象データを生成する。ここで、アクセス要求を受信してからアクセス対象データを生成するまでの処理時間は、アクセス要求に対応するサーバプログラムの相違や、サーバ装置 110 を実現するコンピュータの相違等の要因により異なる。そこで、本実施形態に係るキャッシュ最適化装置 140 は、複数のアクセス対象データの少なくとも1つについて、キャッシュミス時のサーバ装置 110 の処理時間を示すアクセスコストが、他のアクセス対象データと異なる場合において、キャッシュ装置 130 のアクセス処理性能を最適化することを目的とする。

#### 【0023】

図 2 は、本実施形態に係るキャッシュ最適化装置 140 の構成を示す。キャッシュ最適化装置 140 は、パラメータ設定部 200 と、グループ対応統計情報取得部 210 と、アクセス要求到着頻度取得部 215 と、無効化要求到着頻度取得部 220 と、アクセス要求到着確率密度関数生成部 225 と、無効化要求到着確率密度関数生成部 230 と、次アクセスヒット時間分布関数生成部 235 と、キャッシュ有効時間分布関数生成部 240 と、キャッシュヒット率推定関数生成部 245 と、アクセス処理性能推定関数生成部 250 と、パラメータ変更部 255 とを備える。

#### 【0024】

パラメータ設定部 200 は、キャッシュの設定パラメータの初期値をキャッシュ装置 130 に設定する。ここで、パラメータ設定部 200 が設定する設定パラメータは、要求元装置 100 からのアクセス要求に対するアクセス処理性能を変化させるパラメータである。

#### 【0025】

グループ対応統計情報取得部 210 は、アクセス要求及び/又はアクセス応答についてキャッシュ装置 130 により計測された統計情報をキャッシュ装置 130 から取得する。本実施形態において、キャッシュ装置 130 にキャッシュされる各アクセス対象データは、アクセスコスト等に基づいて分類した複数のグループのいずれかに属する。一例として、アクセス要求に応じてサーバ装置 110 により生成されるアクセス対象データは、当該アクセス対象データを生成するサーバプログラム毎、すなわち例えばサーバレット、コマンド、又は Web サービスの種類毎にグループ化される。この場合、異なるサーバプログラムに対する複数のアクセス要求に対して生成される複数のアクセス対象データは、異なるグループに属する。一方、同一のサーバプログラムへ入力パラメータが異なる複数のアクセス要求が送信された場合に生成される複数のアクセス対象データは、同一のグループに属する。後者の具体的な例としては、ユーザ ID の異なる複数の使用者が、同一のサーバプログラムに対してアクセスする場合等が挙げられる。

本実施形態において、グループ対応統計情報取得部 210 は、アクセス要求及びアクセス対象データのグループ毎に、キャッシュ装置 130 により計測された統計情報を取得する。

#### 【0026】

アクセス要求到着頻度取得部 215 は、グループ対応統計情報取得部 210 により取得

10

20

30

40

50

された統計情報に基づいて、それぞれのアクセス対象データに対するアクセス要求について計測された平均到着頻度  $\mu_i$  を取得する。無効化要求到着頻度取得部 220 は、グループ対応統計情報取得部 210 により取得された統計情報、及び平均到着頻度  $\mu_i$  に基づいて、それぞれのアクセス対象データに対する無効化要求について計測された平均到着頻度  $\mu_i$  を取得する。

#### 【0027】

アクセス要求到着確率密度関数生成部 225 は、それぞれのアクセス対象データに対するアクセス要求の平均到着頻度  $\mu_i$  に基づいて、当該アクセス対象データに対するアクセス要求の到着時間間隔の確率密度関数であるアクセス要求到着確率密度関数  $f_i^R(t)$  を生成する。無効化要求到着確率密度関数生成部 230 は、それぞれのアクセス対象データに対する無効化要求の平均到着頻度  $\mu_i$  に基づいて、当該アクセス対象データに対する無効化要求の到着時間間隔の確率密度関数である無効化要求到着確率密度関数  $f_i^I(t)$  を生成する。

10

#### 【0028】

次アクセスヒット時間分布関数生成部 235 は、それぞれのアクセス対象データについて、アクセス要求到着確率密度関数  $f_i^R(t)$  及び無効化要求到着確率密度関数  $f_i^I(t)$  に基づいて、次アクセスヒット時間分布関数  $F_i(t)$  を生成する。本実施形態において、次アクセスヒット時間分布関数は、当該アクセス対象データへのアクセス要求を受信した後、当該アクセス対象データがキャッシュから無効化される前に当該アクセス対象データに対する次のアクセス要求を受信する確率の時間分布関数である。

20

#### 【0029】

キャッシュ有効時間分布関数生成部 240 は、それぞれのアクセス対象データについて、アクセス要求到着確率密度関数  $f_i^R(t)$  及び無効化要求到着確率密度関数  $f_i^I(t)$  に基づいて、キャッシュ有効時間分布関数  $G_i(t)$  を生成する。本実施形態において、キャッシュ有効時間分布関数  $G_i(t)$  は、予め定められた時点、すなわち例えば任意の時点を基準とし、当該アクセス対象データに対する少なくとも1つのアクセス要求を受信し、かつ、当該アクセス対象データがキャッシュから無効化されていない確率の時間分布関数である。

#### 【0030】

キャッシュヒット率推定関数生成部 245 は、複数のアクセス対象データのアクセス要求到着確率密度関数  $f_i^R(t)$  に基づいて、それぞれのアクセス対象データのキャッシュヒット率の推定関数  $h_i$  を生成する。より具体的には、本実施形態に係るキャッシュヒット率推定関数生成部 245 は、それぞれのアクセス対象データについて、アクセス要求到着確率密度関数  $f_i^R(t)$  に基づく複数のアクセス対象データのキャッシュ有効時間分布関数  $G_i(t)$  と、当該アクセス対象データに対する次アクセスヒット時間分布関数  $F_i(t)$  とに基づいて、当該アクセス対象データのキャッシュヒット率推定関数  $h_i$  を生成する。

30

#### 【0031】

アクセス処理性能推定関数生成部 250 は、それぞれのアクセス対象データのキャッシュヒット率推定関数  $h_i$ 、キャッシュヒット時のアクセスコスト  $C_i^R$ 、及びキャッシュミス時のアクセスコスト  $C_i^C$  に基づいて、アクセス処理性能の推定関数  $C$  を生成する。パラメータ変更部 255 は、キャッシュ装置 130 に設定された設定パラメータを、アクセス処理性能推定関数  $C$  によるアクセス処理性能の推定値をより高める設定パラメータに変更する。

40

#### 【0032】

図3は、本実施形態に係るグループ対応統計情報取得部 210 の構成を示す。グループ対応統計情報取得部 210 は、アクセス対象データの各グループ  $i$  ( $i=1, 2, \dots, M$ ) の統計情報として、グループ対応アクセス要求回数  $requestFromClients_i$ 、グループ対応キャッシュヒット回数  $hitsInMemory_i$ 、グループ対応エントリ数  $entries_i$ 、及びグループ対応無効化回数  $explicitInvalidation_i$  をキャッシュ装置 130 から取得する。 $requestFromClients_i$  は、1又は複数の要求元装置 100 からグループ  $i$  のアクセス対象データに対するアクセス要求を受信した回数の計測値である。 $hitsInMemory_i$  は、アクセス要求に対して

50

、グループ*i*のアクセス対象データがキャッシュヒットした回数の計測値である。entries<sub>*i*</sub>は、グループ*i*に属する全てのアクセス対象データのうち、キャッシュ装置130がキャッシュしているアクセス対象データの個数の計測値である。explicitInvalidation<sub>*i*</sub>は、グループ*i*のアクセス対象データをキャッシュから無効化した回数の計測値である。なお、requestFromClients<sub>*i*</sub>、hitsInMemory<sub>*i*</sub>及びexplicitInvalidation<sub>*i*</sub>は、単位時間当たりのカウント値であってよく、entries<sub>*i*</sub>は、単位時間当たりの平均値又は単位時間毎のサンプル値であってよい。グループ対応統計情報取得部210は、グループ対応キャッシュヒット率取得部300と、グループ対応エントリ数取得部310と、アクセス対象データ数算出部320と、グループ対応無効化数取得部330とを有し、これらの計測値に基づいて、アクセス対象データ毎の統計値を取得する。

10

## 【0033】

グループ対応キャッシュヒット率取得部300は、以下の式(4)に示すように、それぞれのグループについて、グループ対応キャッシュヒット回数hitsInMemory<sub>*i*</sub>及びグループ対応アクセス要求回数requestFromClients<sub>*i*</sub>に基づいて、当該グループに属する全てのアクセス対象データについて計測された平均キャッシュヒット率H<sub>*i*</sub>を取得する。

## 【数4】

$$H_i = \text{hitsInMemory}_i / \text{requestFromClients}_i \quad (4)$$

ここで、同一グループに属する各アクセス対象データが同一のアクセス特性を有することを前提とすれば、同一グループに属する複数のアクセス対象データについては、同一の頻度でアクセスされる。このため、グループ*i*の全アクセス対象データのキャッシュヒット率をH<sub>*i*</sub>と見なすことができる。

20

## 【0034】

グループ対応エントリ数取得部310は、それぞれのグループについて、グループ対応エントリ数entries<sub>*i*</sub>の計測値を取得する。アクセス対象データ数算出部320は、それぞれのグループについて、当該グループのグループ対応エントリ数entries<sub>*i*</sub>及び当該グループの平均キャッシュヒット率H<sub>*i*</sub>に基づいて、当該グループに属するアクセス対象データの数*v<sub>i</sub>*の推定値を算出する。ここで、同一グループに属する複数のアクセス対象データは、同一の頻度でアクセスされることから、アクセス対象データ数算出部320は、以下の式(5)に示すように、グループ対応エントリ数entries<sub>*i*</sub>を平均キャッシュヒット率H<sub>*i*</sub>で割ることにより、各グループに属するアクセス対象データの数*v<sub>i</sub>*の推定値を算出することができる。

30

## 【数5】

$$v_i = \text{entries}_i / H_i \quad (5)$$

## 【0035】

グループ対応無効化数取得部330は、それぞれのグループについて、グループ対応無効化回数explicitInvalidation<sub>*i*</sub>の計測値を取得する。本実施形態に係るキャッシュ装置130は、アクセス対象データに変更が生じた場合に、当該アクセス対象データをキャッシュから無効化して、再度サーバ装置110により生成させる。そこでキャッシュ装置130は、アクセス対象データに変更が生じた場合に、要求元装置100又はサーバ装置110から無効化要求を受けて、当該アクセス対象データを無効化する。グループ対応無効化数取得部330は、無効化要求によりアクセス対象データが無効化された回数を、グループ毎に計測する。

40

## 【0036】

以上に示したグループ対応統計情報取得部210によれば、アクセス対象データを適切にグループ化しておくことにより、グループ毎の統計情報に基づいて、個々のアクセス対象データについての統計情報を近似的に求めることができる。このため、キャッシュ装置130は、グループ化した複数のアクセス対象データについての統計情報を計測することができ、個々のアクセス対象データの到着頻度が低い場合においても適切な統計値を得る

50

ことができる。

【0037】

図4は、本実施形態に係るキャッシュ最適化装置140の動作フローを示す。

まず、パラメータ設定部200は、キャッシュの設定パラメータの初期値をキャッシュ装置130に設定する(ステップS400)。この初期値は、情報システム10の管理者等により予め定められてよい。情報システム10は、この設定に基づいて、アプリケーション処理を行う。本実施形態におけるキャッシュの設定パラメータは、一例として、それぞれのアクセス対象データをキャッシュするか否かをグループ毎に指定するパラメータである。

【0038】

次に、グループ対応統計情報取得部210は、アクセス要求及び/又はアクセス応答についてキャッシュ装置130により計測された統計情報をキャッシュ装置130から取得する(S410)。そして、取得したグループ対応アクセス要求回数 $requestFromClients_i$ 及びアクセス対象データ数算出部320により算出したアクセス対象データ数 $v_i$ をアクセス要求到着頻度取得部215に供給する。また、取得したグループ対応キャッシュヒット回数 $hitsInMemory_i$ 及びグループ対応無効化回数 $explicitInvalidation_i$ を無効化要求到着頻度取得部220に供給する。

【0039】

次に、アクセス要求到着頻度取得部215は、グループ対応統計情報取得部210により取得された統計情報に基づいて、それぞれのアクセス対象データに対するアクセス要求についての平均到着頻度 $\lambda_i$ の計測値を取得する(S420)。本実施形態において、アクセス要求到着頻度取得部215は、以下の式(6)に示すように、それぞれのグループについて、グループ対応アクセス要求回数 $requestFromClients_i$ を当該グループのアクセス対象データ数 $v_i$ で割ることにより、各アクセス対象データに対するアクセス要求の平均到着頻度 $\lambda_i$ を取得する。ここで、グループ対応アクセス要求回数 $requestFromClients_i$ は、当該グループに属する全アクセス対象データについて計測された単位時間当たりのアクセス要求数、すなわち、アクセス要求の平均到着頻度である。

【数6】

$$\lambda_i = requestFromClients_i / v_i \quad (6)$$

【0040】

次に、無効化要求到着頻度取得部220は、グループ対応統計情報取得部210により取得された統計情報に基づいて、それぞれのアクセス対象データに対する無効化要求の平均到着頻度 $\mu_i$ の計測値を取得する(S430)。本実施形態において、無効化要求到着頻度取得部220は、以下の式(7)に示すように、それぞれのグループについて、アクセス要求到着頻度取得部215により取得されたアクセス要求の平均到着頻度 $\lambda_i$ と、グループ対応統計情報取得部210により取得されたグループ対応無効化回数 $explicitInvalidation_i$ 、及びグループ対応キャッシュヒット回数 $hitsInMemory_i$ とに基づいて、各アクセス対象データに対する無効化要求の平均到着頻度 $\mu_i$ を取得する。

【数7】

$$\mu_i = \lambda_i \cdot explicitInvalidation_i / hitsInMemory_i \quad (7)$$

【0041】

次に、アクセス要求到着確率密度関数生成部225は、それぞれのアクセス対象データに対するアクセス要求の平均到着頻度 $\lambda_i$ に基づいて、当該アクセス対象データに対するアクセス要求到着確率密度関数 $f_i^R(t)$ を生成する(S440)。次に、無効化要求到着確率密度関数生成部230は、それぞれのアクセス対象データに対する無効化要求の平均到着頻度 $\mu_i$ に基づいて、当該アクセス対象データに対するアクセス要求到着確率密度関数 $f_i^I(t)$ を生成する(S450)。

10

20

30

40

50

## 【 0 0 4 2 】

次に、次アクセスヒット時間分布関数生成部 2 3 5 は、それぞれのアクセス対象データについて、アクセス要求到着確率密度関数  $f_i^R(t)$  及び無効化要求到着確率密度関数  $f_i^I(t)$  に基づいて、次アクセスヒット時間分布関数  $F_i(t)$  を生成する ( S 4 5 2 )。より具体的には、次アクセスヒット時間分布関数生成部 2 3 5 は、次の ( a ) から ( d ) に示す手順により次アクセスヒット時間分布関数  $F_i(t)$  を生成する。

## 【 0 0 4 3 】

( a ) 当該アクセス対象データへの無効化要求を受信した後、次に当該アクセス対象データへの無効化要求を受信するまでの時間分布関数  $F_i^I(t)$  を、無効化要求到着確率密度関数  $f_i^I(t)$  に基づいて生成 ( 式 ( 8 ) )

## 【 数 8 】

$$F_i^I(t) = \int_0^t f_i^I(s) ds \quad (8)$$

## 【 0 0 4 4 】

( b ) 任意の時点を基準として、次の無効化要求が到着するまでの時間 ( 前方待ち時間 ) 又は最後に無効化要求が到着してからの経過時間 ( 後方経過時間 ) の時間分布関数  $G_i^I(t)$  を、無効化要求到着確率密度関数  $f_i^I(t)$  及び時間分布関数  $F_i^I(t)$  に基づいて生成 ( 式 ( 9 ) )

## 【 数 9 】

$$G_i^I(t) = \frac{1}{m_i^R} \int_0^t \left( 1 - \int_0^s f_i^I(u) du \right) ds = \frac{1}{m_i^I} \int_0^t (1 - F_i^I(s)) ds \quad \left( m_i^I = \int_0^\infty s f_i^I(s) ds \right) \quad (9)$$

## 【 0 0 4 5 】

( c ) 任意の時点を基準として、ある時刻  $t$  までに無効化要求が到着していない確率  $\alpha_i(t)$  を、時間分布関数  $G_i^I(t)$  に基づいて生成 ( 式 ( 1 0 ) )

## 【 数 1 0 】

$$\alpha_i(t) = 1 - G_i^I(t) \quad (10)$$

## 【 0 0 4 6 】

( d ) 確率  $\alpha_i(t)$  及びアクセス要求到着確率密度関数  $f_i^R(t)$  に基づいて、次アクセスヒット時間分布関数  $F_i(t)$  を生成 ( 式 ( 1 1 ) )

## 【 数 1 1 】

$$F_i(t) = \int_0^t \alpha_i(s) f_i^R(s) ds \quad (11)$$

## 【 0 0 4 7 】

以上において、次アクセスヒット時間分布関数生成部 2 3 5 は、それぞれのアクセス対象データをキャッシュするか否かを設定する設定パラメータを入力とする次アクセスヒット時間分布関数  $F_i(t)$  を生成する。すなわち例えば、次アクセスヒット時間分布関数生成部 2 3 5 は、グループ  $i$  のアクセス対象データをキャッシュしない場合に、グループ  $i$  のアクセス要求到着確率密度関数  $f_i^R(t)$  及び無効化要求到着確率密度関数  $f_i^I(t)$  を "0" として上記の計算を行う次アクセスヒット時間分布関数  $F_i(t)$  を生成する。

## 【 0 0 4 8 】

次に、キャッシュ有効時間分布関数生成部 2 4 0 は、それぞれのアクセス対象データについて、アクセス要求到着確率密度関数  $f_i^R(t)$  及び無効化要求到着確率密度関数  $f_i^I(t)$  に基づいて、キャッシュ有効時間分布関数  $G_i(t)$  を生成する ( S 4 5 4 )。より具体的には、キャッシュ有効時間分布関数生成部 2 4 0 は、次の ( a ) から ( b ) に示す手順によりキャッシュ有効時間分布関数  $G_i(t)$  を生成する。

10

20

30

40

50

## 【 0 0 4 9 】

( a ) 当該アクセス対象データへのアクセス要求を受信した後、次に当該アクセス対象データへのアクセス要求を受信するまでの時間分布関数  $F_i^R(t)$  を、アクセス要求到着確率密度関数  $f_i^R(t)$  に基づいて生成 ( 式 ( 1 2 ) )

## 【 数 1 2 】

$$F_i^R(t) = \int_0^t f_i^R(s) ds \quad (12)$$

## 【 0 0 5 0 】

( b ) 次アクセスヒット時間分布関数生成部 2 3 5 が生成した確率  $\alpha_i(t)$  と、時間分布関数  $F_i^R(t)$  及びアクセス要求到着確率密度関数  $f_i^R(t)$  に基づいて、キャッシュ有効時間分布関数  $G_i(t)$  を生成

キャッシュ有効時間分布関数  $G_i(t)$  は、時刻 0 から  $t$  の間にアクセス要求が到着し、アクセス要求の到着後時刻  $t$  までに無効化要求が到着していない確率であるから、以下の式 ( 1 3 ) により求めることができる。

## 【 数 1 3 】

$$\begin{aligned} G_i(t) &= \frac{1}{m_i^R} \int_0^t \alpha_i(t-s) (1 - F_i^R(t-s)) ds \\ &= \frac{1}{m_i^R} \int_0^t \alpha_i(s) (1 - F_i^R(s)) ds \quad \left( m_i^R = \int_0^\infty s f_i^R(s) ds \right) \end{aligned} \quad (13)$$

## 【 0 0 5 1 】

以上において、キャッシュ有効時間分布関数生成部 2 4 0 は、それぞれのアクセス対象データをキャッシュするか否かを設定する設定パラメータを入力とするキャッシュ有効時間分布関数  $G_i(t)$  を生成する。すなわち例えば、キャッシュ有効時間分布関数生成部 2 4 0 は、グループ  $i$  のアクセス対象データをキャッシュしない場合に、グループ  $i$  のアクセス要求到着確率密度関数  $f_i^R(t)$  及び無効化要求到着確率密度関数  $f_i^I(t)$  を "0" として上記の計算を行うキャッシュ有効時間分布関数  $G_i(t)$  を生成する。

## 【 0 0 5 2 】

次に、キャッシュヒット率推定関数生成部 2 4 5 は、それぞれのグループに属するアクセス対象データについて、複数のアクセス対象データのキャッシュ有効時間分布関数  $G_i(t)$  と、当該アクセス対象データに対する次アクセスヒット時間分布関数  $F_i(t)$  とに基づいて、当該アクセス対象データのキャッシュヒット率推定関数  $h_i$  を生成する ( S 4 6 0 )。ここで、次アクセスヒット時間分布関数  $F_i(t)$  及びキャッシュ有効時間分布関数  $G_i(t)$  は設定パラメータを入力とする関数であるから、キャッシュヒット率推定関数生成部 2 4 5 は、当該設定パラメータを入力とするキャッシュヒット率推定関数  $h_i$  を生成することができる。

## 【 0 0 5 3 】

次に、アクセス処理性能推定関数生成部 2 5 0 は、それぞれのアクセス対象データのキャッシュヒット率推定関数  $h_i$ 、キャッシュヒット時のアクセスコスト  $C_i^R$ 、及びキャッシュミス時のアクセスコスト  $C_i^C$  に基づいて、アクセス処理性能の推定関数  $C$  を生成する ( S 4 7 0 )。より具体的には、アクセス処理性能推定関数生成部 2 5 0 は、グループ対応統計情報取得部 2 1 0 及びアクセス要求到着頻度取得部 2 1 5 により算出されたグループ  $i$  のアクセス対象データ数  $N_i$  及びアクセス要求の平均到着頻度  $\lambda_i$  と、アクセス対象データのキャッシュヒット率推定関数  $h_i$  と、キャッシュヒット時及びミス時のアクセスコスト  $C_i^R$  及び  $C_i^C$  とに基づいて、以下の式 ( 1 4 ) によりアクセス処理性能の推定関数  $C$  を生成する。

【数 1 4】

$$C = \sum_{i=1}^M v_i \lambda_i (h_i C_i^r + (1-h_i) C_i^c) \quad (14)$$

【0054】

ここで、キャッシュヒット率推定関数  $h_i$  は設定パラメータを入力とする関数であるから、アクセス処理性能推定関数生成部 250 は、当該設定パラメータを入力とするアクセス処理性能推定関数  $C$  を生成することができる。

【0055】

次に、パラメータ変更部 255 は、キャッシュ装置 130 に設定された設定パラメータを、アクセス処理性能推定関数によるアクセス処理性能の推定値をより高める設定パラメータに変更する (S480)。本実施形態に係るパラメータ変更部 255 は、以下の (a) から (d) に示す近傍探索法によりアクセス処理性能の推定値を高める設定パラメータを求め、キャッシュ装置 130 に設定する。

【0056】

(a) 現在の設定パラメータについてのアクセス処理性能の推定値を算出する。  
 (b) 最適パラメータを空集合に初期化する。  
 (c) 最適パラメータにおける各グループの設定パラメータについて、当該グループに属するアクセス対象データをキャッシュする場合及びキャッシュしない場合のアクセス処理性能の推定値を算出する。そして、当該アクセス対象データをキャッシュする場合に、キャッシュしない場合と比較してアクセス処理性能の推定値がより高くなることを条件として、当該グループに属するアクセス対象データをキャッシュする設定パラメータを最適パラメータに追加する。一方、当該アクセス対象データをキャッシュする場合に、キャッシュしない場合と比較してアクセス処理性能の推定値がより高くないこと条件として、当該グループに属するアクセス対象データをキャッシュしない設定パラメータを最適パラメータに追加する。  
 (d) 全てのグループについて (c) を行った結果得られた最適パラメータによるアクセス処理性能の推定値が、現在の設定パラメータについてのアクセス処理性能の推定値より高い場合に、最適パラメータをキャッシュ装置 130 に設定する。

【0057】

なお、以上に示した (c) の処理に代えて、パラメータ変更部 255 は、以下に示す処理を行ってもよい。

(c') 現在の設定パラメータにおける各グループの設定パラメータについて、当該グループに属するアクセス対象データをキャッシュする設定となっている場合に、当該アクセス対象データをキャッシュしない場合のアクセス処理性能の推定値を算出する。そして、当該アクセス対象データをキャッシュしない場合に、最適パラメータと比較してアクセス処理性能の推定値が高くなることを条件として、現在の設定パラメータにおいて当該グループに属するアクセス対象データをキャッシュしないように変更した設定パラメータを最適パラメータとする。

一方、当該グループに属するアクセス対象データをキャッシュしない設定となっている場合に、当該アクセス対象データをキャッシュする場合のアクセス処理性能の推定値を算出する。そして、当該アクセス対象データをキャッシュする場合に、最適パラメータと比較してアクセス処理性能の推定値が高くなることを条件として、現在の設定パラメータにおいて当該グループに属するアクセス対象データをキャッシュする変更した設定パラメータを最適パラメータとする。

【0058】

キャッシュ最適化装置 140 は、以上の S410 から S480 に示した処理を、例えば定期的に繰り返し行うことにより、キャッシュ装置 130 の設定パラメータを最適値に保つ (S490)。

10

20

30

40

50



## 【 0 0 5 9 】

以上の処理により、パラメータ変更部 2 5 5 は、アクセス対象データをキャッシュする場合に、キャッシュしない場合と比較してアクセス処理性能の推定値がより高くなることを条件として、当該アクセス対象データをキャッシュすることを指定する設定パラメータをキャッシュ装置 1 3 0 に設定することができる。そして、パラメータ変更部 2 5 5 は、アクセス処理性能推定関数 C に基づいて、全てのグループについてキャッシュ有無を指定する最適パラメータを予め求めてから、キャッシュ装置 1 3 0 に設定することができる。この結果、キャッシュ最適化装置 1 4 0 は、各グループについてキャッシュ有無を試行する場合と比較して、より効率良くキャッシュ装置 1 3 0 のアクセス処理性能を最適化することができる。

10

## 【 0 0 6 0 】

なお、キャッシュ装置 1 3 0 が、アクセス対象データをタイムアウト時に無効化する方式を採用する場合、式 ( 1 1 ) 及び式 ( 1 3 ) に示した次アクセスヒット時間分布関数  $F_i(t)$  及びキャッシュ有効時間分布関数  $G_i(t)$  に代えて、以下に示す時間分布関数を用いることもできる。

## 【 0 0 6 1 】

この方式においては、キャッシュ装置 1 3 0 は、キャッシュしているアクセス対象データに対して無効化要求を受信した場合、又は、当該アクセス対象データをキャッシュしてから当該アクセス対象データに対してアクセス要求を受けることなく予め定められたタイムアウト時間  $\tau_i$  が経過した場合に、当該アクセス対象データをキャッシュから無効化する。このため、アクセス要求が到着してからタイムアウト時間  $\tau_i$  の経過後は、次アクセスヒット時間分布関数  $F_i(t)$  及びキャッシュ有効時間分布関数  $G_i(t)$  が一定となる。そこで、次アクセスヒット時間分布関数生成部 2 3 5 及びキャッシュ有効時間分布関数生成部 2 4 0 は、予め定められたタイムアウト時間  $\tau_i$  の経過前及び経過後の各区間について、異なる計算式を用いた次アクセスヒット時間分布関数  $F_i(t)$  及びキャッシュ有効時間分布関数  $G_i(t)$  を生成する ( 式 ( 1 5 ) 及び式 ( 1 6 ) ) 。

20

## 【 数 1 5 】

$$F_i(t) = \begin{cases} \int_0^t \alpha_i(s) f_i^R(s) ds & (t \leq \tau_i) \\ \int_0^{\tau_i} \alpha_i(s) f_i^R(s) ds & (t > \tau_i) \end{cases} \quad (15)$$

30

## 【 0 0 6 2 】

## 【 数 1 6 】

$$G_i(t) = \begin{cases} \frac{1}{m_i^R} \int_0^t \alpha_i(s) (1 - F_i^R(s)) ds & (t \leq \tau_i) \\ \frac{1}{m_i^R} \int_0^{\tau_i} \alpha_i(s) (1 - F_i^R(s)) ds & (t > \tau_i) \end{cases} \quad (16)$$

40

## 【 0 0 6 3 】

また、以上において、アクセス要求到着確率密度関数  $f_i^R(t)$  及び無効化要求到着確率密度関数  $f_i^I(t)$  が指数分布でない場合、式 ( 1 1 ) 及び ( 1 3 ) 、又は式 ( 1 5 ) 及び ( 1 6 ) に示した積分関数を求めるのが困難となる場合がある。この場合において、次アクセスヒット時間分布関数生成部 2 3 5 及びキャッシュ有効時間分布関数生成部 2 4 0 は、線分で近似したアクセス要求到着確率密度関数  $f_i^R(t)$  及び無効化要求到着確率密度関数  $f_i^I(t)$  に基づいて、次アクセスヒット時間分布関数  $F_i(t)$  及びキャッシュ有効時間分布関数  $G_i(t)$  を生成してもよい。

50

## 【 0 0 6 4 】

図 5 は、本実施形態に係るキャッシュヒット率推定関数生成部 2 4 5 の構成を示す。キャッシュヒット率推定関数生成部 2 4 5 は、エントリ数期待値関数生成部 5 0 0 と、キャッシュ充填時間算出部 5 1 0 と、次アクセスヒット率推定関数生成部 5 2 0 とを有する。

## 【 0 0 6 5 】

エントリ数期待値関数生成部 5 0 0 は、複数のアクセス対象データのキャッシュ有効時間分布関数  $G_i(t)$  に基づいて、エントリ数期待値関数  $G(t)$  を生成する。エントリ数期待値関数  $G(t)$  とは、無効化されることなくキャッシュされているアクセス対象データの数の期待値を求める関数である。ここで、キャッシュ有効時間分布関数  $G_i(t)$  は、グループ  $i$  に属する 1 つのアクセス対象データに関して、当該アクセス対象データへのアクセス要求を受信した時点を基準とし、時刻  $t$  までにアクセス要求が到着し、かつ、時刻  $t$  までに無効化されていない確率である。したがって、エントリ数期待値関数生成部 5 0 0 は、キャッシュ有効時間分布関数  $G_i(t)$  にグループ  $i$  に属するアクセス対象データの数  $v_i$  を乗じて、無効化されることなくキャッシュされているグループ  $i$  のアクセス対象データの数の期待値を求めることができる。そして、エントリ数期待値関数生成部 5 0 0 は、式 ( 1 7 ) に示すように、この期待値を全てのグループについて合計することにより、エントリ数期待値関数  $G(t)$  を求めることができる。

10

## 【 数 1 7 】

$$G(t) = \sum_i v_i G_i(t) \quad (17)$$

20

## 【 0 0 6 6 】

キャッシュ充填時間算出部 5 1 0 は、エントリ数期待値関数  $G(t)$  に基づいて、予め定められた時点を基準として、 $K$  個のアクセス対象データがキャッシュされるまでのキャッシュ充填時間  $T_{fill}$  の期待値を算出する。より具体的には、キャッシュ充填時間算出部 5 1 0 は、エントリ数期待値関数  $G(t)$  の値がキャッシュサイズ  $K$  となるまでの時間の期待値を以下の式 ( 1 8 ) により算出し、キャッシュ充填時間  $T_{fill}$  の期待値とする。

## 【 数 1 8 】

$$k = G(T_{fill}) \Rightarrow T_{fill} = G^{-1}(k) \quad (18)$$

30

## 【 0 0 6 7 】

次アクセスヒット率推定関数生成部 5 2 0 は、それぞれのアクセス対象データについて、当該アクセス対象データに対するアクセス要求を受信してからキャッシュ充填時間  $T_{fill}$  の期待値により定められる時間の経過までに当該アクセス対象データに対する次のアクセス要求を受信する確率の関数を、次アクセスヒット時間分布関数  $F_i(t)$  に基づいて算出し、当該アクセス対象データのキャッシュヒット率推定関数  $h_i$  とする。より具体的には、次アクセスヒット率推定関数生成部 5 2 0 は、キャッシュ充填時間  $T_{fill}$  が経過するまでに、注目するアクセス対象データへのアクセス要求が到着し、当該アクセス対象データが無効化されていない確率を、以下の式 ( 1 9 ) により算出し、キャッシュヒット率推定関数  $h_i$  とする。

40

## 【 数 1 9 】

$$h_i = F_i(G^{-1}(K)) \quad (19)$$

## 【 0 0 6 8 】

以上に示したキャッシュ最適化装置 1 4 0 によれば、LRU方式のキャッシュ装置 1 3 0 について、キャッシュミス時のアクセスコストがグループ間で異なる場合においても、各グループに属するアクセス対象データのキャッシュヒット率の推定値を解析的に求めることができる。そして、キャッシュ最適化装置 1 4 0 は、各グループについてのキャッシュヒット率に基づいて、アクセス処理性能の推定関数を精度良く求めることができる。この結果キャッシュ最適化装置 1 4 0 は、アクセス処理性能推定関数を用いて最適な設定パ

50

ラメータを精度良く求めることができ、効率良くキャッシュ装置 130 のアクセス処理性能を高めることができる。

【0069】

なお、アクセス対象データのタイムアウトを考慮した場合、時刻  $T_{f_{i+1}}$  においてキャッシュされているアクセス対象データの数が  $K$  個であっても、時刻  $T_{f_{i+1}}$  以前に  $K$  個を超えるアクセス対象データがキャッシュされた後タイムアウトにより無効化されたケースが考えられる。このようなケースにおいては、アクセス対象データがリプレイスされるため、キャッシュミスとなる。したがって、キャッシュ最適化装置 140 は、上記の式 (17) から (19) を満たすキャッシュヒット率推定関数  $h_i$  を、キャッシュヒット率の上界を与える近似式として用いる。

【0070】

また、アクセス対象データのタイムアウトを考慮した場合、次アクセスヒット時間分布関数  $F_i(t)$  及びキャッシュ有効時間分布関数  $G_i(t)$  は、時刻  $t_i$  で不連続となる。そこで、エントリ数期待値関数生成部 500 は、各アクセス対象データをタイムアウト時間  $t_i$  の値でソートし、複数のタイムアウト時間  $t_i$  で区切られた各区間毎に、当該区間におけるキャッシュ充填時間  $T_{f_{i+1}}$  の期待値を算出してもよい。

【0071】

図 6 は、本実施形態の第 1 変形例に係るキャッシュ最適化装置 140 によるアクセス処理性能最適化方法の一例を示す。本変形例において、キャッシュ装置 130 は、それぞれのアクセス対象データ 600 を、キャッシュ装置 130 のキャッシュ記憶領域 135 を分割した複数のキャッシュ分割領域 610 のうち、当該アクセス対象データに対応して予め定められたキャッシュ分割領域 610 にキャッシュする。

【0072】

図 6 (a) はアクセス対象データ 600 の配置変更によるアクセス処理性能最適化を示す。図 6 (a) において、キャッシュ装置 130 は、それぞれのアクセス対象データ 600 を、設定パラメータにより定められた、当該アクセス対象データに対応するキャッシュ分割領域 610 にキャッシュする。そして、キャッシュ最適化装置 140 は、設定パラメータを変更して、それぞれのアクセス対象データ 600 を、いずれのキャッシュ分割領域 610 にキャッシュするかを最適化することにより、キャッシュ装置 130 のアクセス処理性能を高める。

【0073】

より具体的には、まずパラメータ設定部 200 は、それぞれのアクセス対象データ 600 を、いずれのキャッシュ分割領域 610 にキャッシュするかを設定パラメータの初期値によりキャッシュ装置 130 に設定する。ここでパラメータ設定部 200 は、少なくとも 1 つのアクセス対象データ 600 を、キャッシュしないことを設定してもよい。

【0074】

次に、キャッシュヒット率推定関数生成部 245 及びアクセス処理性能推定関数生成部 250 は、それぞれのアクセス対象データ 600 をいずれのキャッシュ分割領域 610 にキャッシュするか、又は、キャッシュしないかの対応を指定する組み合わせを入力とするキャッシュヒット率推定関数  $h_i$  及びアクセス処理性能推定関数  $C$  を生成する。例えば、キャッシュヒット率推定関数生成部 245 及びアクセス処理性能推定関数生成部 250 は、キャッシュ分割領域 610 毎に、当該キャッシュ分割領域 610 にキャッシュされる各アクセス対象データについてのヒット率推定関数及びアクセス処理性能推定関数を生成し、キャッシュ分割領域 610 毎のアクセス処理性能推定関数を合計することにより、キャッシュ装置 130 のアクセス処理性能推定関数  $C$  を生成する。

【0075】

そして、パラメータ変更部 255 は、それぞれのアクセス対象データ 600 をいずれのキャッシュ分割領域 610 にキャッシュするか、又は、キャッシュしないかの対応を第 1 の組み合わせとした場合に、第 2 の組み合わせとした場合と比較してアクセス処理性能の推定値がより高くなることを条件として、当該対応を第 1 の組み合わせとする設定パラメ

10

20

30

40

50

ータをキャッシュ装置 130 に設定する。より具体的には、パラメータ変更部 255 は、図 4 の S480 に関連して示した近傍探索法と同様にして、それぞれのアクセス対象データ 600 をいずれのキャッシュ分割領域 610 にキャッシュするか、又は、キャッシュしないかの対応の最適パラメータを探索し、キャッシュ装置 130 に設定する。

【0076】

図 6 (b) はキャッシュ分割領域 610 のサイズ変更によるアクセス処理性能最適化を示す。図 6 (b) において、キャッシュ最適化装置 140 は、設定パラメータを変更して、予めアクセス対象データ 600 の割り当てが定められた複数のキャッシュ分割領域 610 のそれぞれについて、キャッシュ記憶領域 135 に対する当該キャッシュ分割領域 610 の割合を最適化することにより、キャッシュ装置 130 のアクセス処理性能を高める。

10

【0077】

より具体的には、キャッシュヒット率推定関数生成部 245 及びアクセス処理性能推定関数生成部 250 は、キャッシュ記憶領域 135 に対する複数のキャッシュ分割領域 610 の割合の組み合わせを入力とするキャッシュヒット率推定関数  $h_i$  及び前記アクセス処理性能推定関数  $C$  を生成する。例えばキャッシュヒット率推定関数生成部 245 及びアクセス処理性能推定関数生成部 250 は、キャッシュ分割領域 610 毎に、キャッシュできるアクセス対象データの個数  $K$  を入力とする、当該キャッシュ分割領域 610 にキャッシュされる各アクセス対象データについてのヒット率推定関数及びアクセス処理性能推定関数を生成する。そして、キャッシュ分割領域 610 毎のアクセス処理性能推定関数を合計することにより、キャッシュ装置 130 のアクセス処理性能推定関数  $C$  を生成する。

20

【0078】

そして、パラメータ変更部 255 は、複数のキャッシュ分割領域 610 の割合の組み合わせを第 1 の組み合わせとした場合に、第 2 の組み合わせとした場合と比較してアクセス処理性能の推定値がより高くなることを条件として、複数のキャッシュ分割領域 610 の割合の組み合わせを第 1 の組み合わせとする設定パラメータをキャッシュ装置 130 に設定する。より具体的には、パラメータ変更部 255 は、最急勾配法を用いてアクセス処理性能の推定値を最大とするキャッシュ分割領域 610 の割合の組み合わせを求め、キャッシュ装置 130 に設定する。

【0079】

以上に示した第 1 変形例に係るキャッシュ最適化装置 140 によれば、各アクセス対象データ 600 を複数のキャッシュ分割領域 610 に割り当てる場合において、アクセス対象データ 600 の割り当て及び / 又は各キャッシュ分割領域 610 のサイズを最適化することができる。

30

【0080】

図 7 は、本実施形態の第 2 変形例に係るキャッシュ装置 130 の構成を示す。本変形例に係るキャッシュ装置 130 は、優先度付の LRU キャッシュ方式を採用する。キャッシュ装置 130 は、キャッシュ記憶領域 135 と、アクセス要求処理部 700 と、LRU リスト管理部 710 と、エントリ移動部 730 と、LRU シフト部 740 と、統計情報計測部 750 とを備える。

【0081】

40

アクセス要求処理部 700 は、要求元装置 100 からのアクセス要求を処理する。より具体的には、アクセス要求処理部 700 は、アクセス要求に対応するアクセス対象データ 600 がキャッシュ記憶領域 135 にキャッシュされているか否かを、LRU リスト管理部 710 により管理される情報に基づいて判定する。そして、アクセス要求処理部 700 は、当該アクセス対象データ 600 がキャッシュされている場合に、当該アクセス対象データ 600 をキャッシュ記憶領域 135 から読み出して要求元装置 100 へ返信する。一方、アクセス要求処理部 700 は、当該アクセス対象データ 600 がキャッシュされていない場合に、アクセス要求をサーバ装置 110 へ転送する。そしてアクセス要求処理部 700 は、サーバ装置 110 から返信されたアクセス対象データ 600 を要求元装置 100 へ返信すると共にキャッシュ記憶領域 135 に登録する。

50

## 【0082】

L R Uリスト管理部710は、キャッシュ記憶領域135に記憶された複数のアクセス対象データ600を管理する。本変形例において、それぞれのアクセス対象データ600には、当該アクセス対象データ600をキャッシュする優先度が定められる。このため、L R Uリスト管理部710は、キャッシュされたアクセス対象データ600をL R U方式により管理する、優先度毎に設けられた複数のL R Uリスト715を有する。各L R Uリスト715は、キャッシュ記憶領域135に記憶されたアクセス対象データ600を特定する1又は複数のポインタ720をリスト構造により管理する。

## 【0083】

10 エントリ移動部730は、アクセス要求を受けたアクセス対象データ600を、当該アクセス対象データ600の優先度に対応するL R Uリスト715の先頭に移動する。これにより、エントリ移動部730は、各アクセス対象データ600の優先度を反映させたL R Uキャッシュを実現することができる。

## 【0084】

L R Uシフト部740は、優先度が最も低いL R Uリスト715が空となった場合に、複数のL R Uリスト715のそれぞれに登録されたアクセス対象データ600を、より優先度が低いL R Uリスト715に登録し直すシフト処理を行う。すなわち例えば、L R Uリスト管理部710が第1から第3L R Uリスト715を有し、第1L R Uリスト715の優先度が最も低い場合において、L R Uシフト部740は、第1L R Uリスト715が空20 となったことを条件として、第2L R Uリスト715に登録されたアクセス対象データ600を第1L R Uリスト715に登録し直し、第3L R Uリスト715に登録されたアクセス対象データ600を第2L R Uリスト715に登録し直す。

## 【0085】

統計情報計測部750は、例えばグループ対応アクセス要求回数 $requestFromClients_i$ 、グループ対応キャッシュヒット回数 $hitsInMemory_i$ 、グループ対応エントリ数 $entries_i$ 、及びグループ対応無効化回数 $explicitInvalidation_i$ 等の、アクセス要求及び/又はアクセス応答についての統計情報を計測し、キャッシュ最適化装置140内のグループ対応統計情報取得部210へ供給する。

## 【0086】

30 以上に示したキャッシュ装置130によれば、より優先度が高いアクセス対象データ600の方が、より優先度が高いL R Uリスト715に登録される。このため、キャッシュ装置130は、各アクセス対象データ600を、優先度にほぼ比例した期間の間キャッシュ記憶領域135に滞在させることができる。

## 【0087】

図8は、本実施形態の第2変形例に係るキャッシュヒット率推定関数生成部245の構成を示す。本変形例に係るキャッシュヒット率推定関数生成部245は、図7に示したキャッシュ装置130について各アクセス対象データのキャッシュヒット率推定関数 $h_i$ を生成することができる。図7に示したキャッシュ装置130においては、各L R Uリスト715に対してシフト処理が行われる度に、不連続に状態が変化する。このため、本変形例に係るキャッシュヒット率推定関数生成部245は、定期的にL R Uリスト715がシフト40 処理が行われるものとして優先度付L R Uキャッシュをモデル化する。

## 【0088】

キャッシュヒット率推定関数生成部245は、エントリ数期待値関数生成部800と、シフト周期算出部810と、リプレース時間算出部820と、次アクセスヒット率推定関数生成部830と、次アクセスヒット率平均化部840とを有する。エントリ数期待値関数生成部800は、複数のアクセス対象データのキャッシュ有効時間分布関数 $G_i(t)$ に基づいて、エントリ数期待値関数 $H(t)$ を生成する。ここで、エントリ数期待値関数 $H(t)$ は、直前のシフト処理を基準として、時刻 $t$ における、複数のL R Uリスト715に登録されたアクセス対象データの合計数の期待値を求める関数である。より具体的には、エントリ数期待値関数生成部800は、グループ毎のキャッシュ有効時間分布関数 $G_i(t)$ と、グル50

ープ毎のアクセス対象データの数  $n_i$  と、グループ毎のアクセス対象データの優先度  $\phi_i$  とに基づいて、以下の式 (20) によりエントリ数期待値関数  $H(t)$  を生成する。

【数 20】

$$H(t) = \sum_{i=1}^M v_i G_i(\phi_i t) \quad (20)$$

【0089】

式 (20) においては、各アクセス対象データは、その優先度  $\phi_i$  の回数分シフトが起きなければキャッシュから掃き出されないことを反映し、キャッシュ有効時間分布関数への入力を  $\phi_i \cdot t$  としている。

【0090】

シフト周期算出部 810 は、エントリ数期待値関数  $H(t)$  に基づいて、シフト処理の周期の期待値  $T_r$  を算出する。より具体的には、シフト周期算出部 810 は、エントリ数期待値関数  $H(t)$  の値がキャッシュサイズ  $K$  となるまでの時間の期待値を、以下の式 (21) により算出し、シフト処理の周期の期待値  $T_r$  とする。

【数 21】

$$T_r = H^{-1}(k) \quad (21)$$

【0091】

リプレース時間算出部 820 は、シフト処理の周期を複数に等分 (例えば  $L$  等分) した各時点において、アクセス対象データへのアクセス要求を受けてから当該アクセス対象データが他のアクセス対象データに置換されるまでのリプレース時間の期待値  $G(t, l)$  を算出する。より具体的には、キャッシュ装置 130 においては、注目するアクセス対象データより優先度が高い他のアクセス対象データがキャッシュされていくことにより、当該アクセス対象データがリプレースされる。ここで、当該アクセス対象データは、シフト処理が行われる度に優先度が低下する結果、当該アクセス対象データのリプレースに寄与する他のアクセス対象データの数が増加していく。これらを反映し、リプレース時間算出部 820 は、グループ  $i$  についてのキャッシュ有効時間分布関数  $G_i(t)$  及び優先度  $\phi_i$  と、グループ  $i$  に属するアクセス対象データの数  $n_i$  と、シフト処理の周期の期待値  $T_r$  とに基づいて、以下の式 (22) によりリプレース時間の期待値  $G(t, l)$  を算出する。

【数 22】

$$G(t, l) = \sum_{\phi_j > \phi_i} v_j G_j \left( t + (\phi_j - \phi_i) T_r + \frac{T_r}{L} l \right) + \sum_{\phi_j = \phi_i} v_j G_j(t) \quad (22)$$

$$+ \sum_{\phi_j < \phi_i} v_j G_j \left( t + (\phi_j - \phi_i - 1) T_r + \frac{T_r}{L} l \right)$$

【0092】

次アクセスヒット率推定関数生成部 830 は、それぞれのアクセス対象データについて、次アクセスヒット時間分布関数  $F_i(t)$  及びリプレース時間  $G(t, l)$  に基づいて、次アクセスヒット率推定関数  $h_{i, j}$  を生成する。ここで、次アクセスヒット率推定関数  $h_{i, j}$  は、シフト処理の周期を  $L$  等分した各時点において当該アクセス対象データに対するアクセス要求を受信してからリプレース時間の期待値により定められる時間  $T_r$  の経過までに、当該アクセス対象データに対する次のアクセス要求を受信する確率である。より具体的には、次アクセスヒット率推定関数生成部 830 は、図 5 に示したキャッシュ充填時間算出部 510 及び次アクセスヒット率推定関数生成部 520 と同様にして、以下の式 (23) により次アクセスヒット率推定関数  $h_{i, j}$  を生成する。

10

20

30

40

【数 2 3】

$$h_{i,l} = F_i(G^{-1}(k,l)) \quad (23)$$

【0093】

次アクセスヒット率平均化部 840 は、それぞれのアクセス対象データについて、シフト処理の周期を複数に等分した複数の時点のそれぞれにおいて当該アクセス対象データに対するアクセス要求を受信した場合の次アクセスヒット率推定関数  $h_{i,l}$  を全ての時点について平均し、当該アクセス対象データのキャッシュヒット率推定関数  $h_i$  とする。すなわち、次アクセスヒット率平均化部 840 は、以下の式 (24) によりキャッシュヒット率推定関数  $h_i$  を生成する。

10

【数 2 4】

$$h_i = \frac{1}{L} \sum_{l=0}^{L-1} h_{i,l} \quad (24)$$

【0094】

本変形例に係るキャッシュ最適化装置 140 によれば、パラメータ変更部 255 は、図 4 の S480 に関して説明した近傍探索法と同様にして、各グループのアクセス対象データをキャッシュする優先度  $\mu_i$  について、アクセス処理性能の推定値を高める設定パラメータを求め、キャッシュ装置 130 に設定する。すなわち、キャッシュヒット率推定関数生成部 245 及びアクセス処理性能推定関数生成部 250 は、それぞれの前記アクセス対象データをキャッシュする優先度を入力とするキャッシュヒット率推定関数  $h_i$  及びアクセス処理性能推定関数  $C$  を生成する。そして、パラメータ変更部 255 は、あるアクセス対象データをキャッシュする優先度を第 1 の優先度とした場合に、第 2 の優先度とした場合と比較してアクセス処理性能の推定値がより高くなることを条件として、当該アクセス対象データをキャッシュする優先度を第 1 の優先度とする設定パラメータをキャッシュ装置 130 に設定する。これにより、キャッシュ最適化装置 140 は、優先度付の LRU キャッシュ方式を採用するキャッシュ装置 130 について、各アクセス対象データをキャッシュする優先度の設定パラメータを最適化することができる。

20

【0095】

図 9 は、本実施形態に係る第 3 変形例に係るキャッシュ最適化装置 140 の構成を示す。図 9 において図 2 と同一符号を付した部材は、図 2 と同様の機能及び構成を備えるため、以下相違点を除き説明を省略する。

30

【0096】

本変形例に係るキャッシュ最適化装置 140 は、複数のキャッシュ装置 130 に対する各アクセス対象データの割り当てを更に最適化する。これを実現するために、キャッシュ最適化装置 140 は、図 2 に示したキャッシュ最適化装置 140 に加え、統合時間関数生成部 920、グループ統合部 930、分割時間関数生成部 940、及びグループ分割部 950 を備える。

【0097】

統合時間関数生成部 920 及びグループ統合部 930 は、複数のキャッシュ装置 130 における同一アクセス特性のグループを統合して、一のキャッシュ装置 130 におけるグループとする処理を行う。すなわち例えば、第 1 のキャッシュ装置 130 として機能する第 1 の JVM と、第 2 のキャッシュ装置 130 として機能する第 2 の JVM とが稼動している場合において、第 1 の JVM のアクセス処理を、第 2 の JVM に担当させるようにアクセス処理の割り当てを変更する。

40

【0098】

統合時間関数生成部 920 は、複数のキャッシュ装置 130 におけるそれぞれのグループを、一のキャッシュ装置 130 における対応するグループに統合した場合におけるアクセス処理性能推定関数  $C$  を生成する。ここで、複数のグループを統合する場合、アクセス要求の総量  $\mu_i$  及び無効化要求の総量  $\mu_i$  は変化しない。一方、同一アクセス特性のグ

50

ループを統合するため、当該グループに属するアクセス対象データの総数  $n_i$  が変化する  
場合と、各アクセス対象データに対するアクセス要求及び無効化要求の平均到着頻度  $\lambda_i$   
及び  $\mu_i$  が変化する場合は有り得る。

【0099】

アクセス対象データの総数  $n_i$  が変化する場合の一例としては、複数のキャッシュ装置  
130が、異なる利用者に対して同一のサービスを提供している場合が挙げられる。この  
ような場合には、アクセス処理を統合すると、各利用者についてのアクセス要求及び無効  
化要求の平均到着頻度は変化しないが、アクセス対象データの総数  $n_i$  は統合前のアクセ  
ス対象データの数の合計となる。そこで統合時関数生成部920は、このケースにおいて  
は、統合後のアクセス要求及び無効化要求の平均到着頻度  $\lambda_i$  及び  $\mu_i$  を統合前の各キャ  
ッシュ装置130における平均到着頻度と同一とする一方、統合後のアクセス対象データの  
数  $n_i$  を、統合前の各キャッシュ装置130におけるアクセス対象データの数の合計とす  
るようグループ対応統計情報取得部210に指示する。この指示を受けると、グループ対  
応統計情報取得部210はアクセス対象データの数を変更し、アクセス要求到着頻度取得  
部215、無効化要求到着頻度取得部220、アクセス要求到着確率密度関数生成部22  
5、無効化要求到着確率密度関数生成部230、次アクセスヒット時間分布関数生成部2  
35、キャッシュ有効時間分布関数生成部240、キャッシュヒット率推定関数生成部2  
45、及びアクセス処理性能推定関数生成部250は統合後のアクセス処理性能推定関数  
Cを生成する。

10

【0100】

各アクセス対象データに対するアクセス要求及び無効化要求の平均到着頻度  $\lambda_i$  及び  $\mu_i$   
が変化する場合の一例としては、複数のキャッシュ装置130が、利用者からのアクセス  
処理に対してサービスを分散処理により提供している場合が挙げられる。このような場合  
には、アクセス処理を統合すると、アクセス対象データの数は変化しないが、各利用者に  
ついてのアクセス要求及び無効化要求の平均到着頻度  $\lambda_i$  及び  $\mu_i$  は統合前の値の合計とな  
る。そこで統合時関数生成部920は、このケースにおいては、統合後のアクセス要求及  
び無効化要求の平均到着頻度  $\lambda_i$  及び  $\mu_i$  を、統合前の各キャッシュ装置130におけるア  
クセス要求及び無効化要求の平均到着頻度の合計とするようアクセス要求到着頻度取得部  
215及び無効化要求到着頻度取得部220に指示する。この指示を受けると、アクセス  
要求到着頻度取得部215及び無効化要求到着頻度取得部220はアクセス要求及び無効  
化要求の平均到着頻度を変更し、アクセス要求到着確率密度関数生成部225、無効化要  
求到着確率密度関数生成部230、次アクセスヒット時間分布関数生成部235、キャッ  
シュ有効時間分布関数生成部240、キャッシュヒット率推定関数生成部245、及びア  
クセス処理性能推定関数生成部250は統合後のアクセス処理性能推定関数Cを生成する  
。

20

30

【0101】

そして、統合時関数生成部920は、上記の指示により変更されたパラメータに基づい  
てアクセス処理性能推定関数生成部250により生成されたアクセス処理性能推定関数C  
を取得することにより、統合後におけるアクセス処理性能推定関数C'を生成する。

【0102】

グループ統合部930は、コンピュータ900により生成されたアクセス処理性能推定  
関数C'によるアクセス処理性能の推定値が、複数のキャッシュ装置130のアクセス処理  
性能の推定値の合計より高い場合に、複数のキャッシュ装置130におけるそれぞれのグ  
ループを、一のキャッシュ装置130における対応するグループに統合する。すなわちグ  
ループ統合部930は、パラメータ変更部255を介してグループの統合をキャッシュ装  
置130に指示する。

40

【0103】

分割時関数生成部940及びグループ分割部950は、一のキャッシュ装置130にお  
けるグループを、複数のキャッシュ装置130に分割する処理を行う。すなわち例えば、  
第1のJVMのアクセス処理を、新たに起動した第2のJVMに一部分担させるようにア

50



クセス処理の割り当てを変更する。

【0104】

分割時間関数生成部940は、一のキャッシュ装置130におけるそれぞれのグループを、複数のキャッシュ装置130における対応するグループに分割した場合における複数のキャッシュ装置130のそれぞれのアクセス処理性能推定関数を生成する。このようにグループを分割した場合においても、アクセス要求の総量  $\mu_i$  及び無効化要求の総量  $\mu_i$  は変化しない。一方、グループを統合する場合と同様に、各グループに属するアクセス対象データの数が変化する場合と、各アクセス対象データに対するアクセス要求及び無効化要求の平均到着頻度が変化する場合が有り得る。

【0105】

アクセス対象データの数を変化させる場合、分割時間関数生成部940は、分割後の各キャッシュ装置130における分割対象グループのアクセス対象データの数の合計値を、分割前のキャッシュ装置130における当該グループのアクセス対象データ数とするようにグループ対応統計情報取得部210に指示し、分割後の各キャッシュ装置130におけるアクセス対象データの数を変更する。この指示を受けると、グループ対応統計情報取得部210は、例えば分割前のキャッシュ装置130におけるアクセス対象データ数を、分割後のキャッシュ装置130の数で割る等の処理を行なって、分割後の各キャッシュ装置130におけるアクセス対象データの数を算出する。そして、アクセス要求到着頻度取得部215、無効化要求到着頻度取得部220、アクセス要求到着確率密度関数生成部225、無効化要求到着確率密度関数生成部230、次アクセスヒット時間分布関数生成部235、キャッシュ有効時間分布関数生成部240、キャッシュヒット率推定関数生成部245、及びアクセス処理性能推定関数生成部250は分割後の各キャッシュ装置130におけるアクセス処理性能推定関数を生成する。

【0106】

一方、アクセス要求及び無効化要求の平均到着頻度を変化させる場合、分割時間関数生成部940は、分割後の各キャッシュ装置130における平均到着頻度の合計値を、分割前のキャッシュ装置130における当該グループの平均到着頻度とするようアクセス要求到着頻度取得部215及び無効化要求到着頻度取得部220に指示し、分割後の各キャッシュ装置130における平均到着頻度を変更する。この指示を受けると、アクセス要求到着頻度取得部215及び無効化要求到着頻度取得部220は、例えば分割前のキャッシュ装置130における平均到着頻度を、分割後のキャッシュ装置130の数で割る等の処理を行なって、分割後の各キャッシュ装置130における平均到着頻度を算出する。そして、アクセス要求到着頻度取得部215、無効化要求到着頻度取得部220、アクセス要求到着確率密度関数生成部225、無効化要求到着確率密度関数生成部230、次アクセスヒット時間分布関数生成部235、キャッシュ有効時間分布関数生成部240、キャッシュヒット率推定関数生成部245、及びアクセス処理性能推定関数生成部250は分割後の各キャッシュ装置130におけるアクセス処理性能推定関数を生成する。

【0107】

そして、分割時間関数生成部940は、上記の指示により変更されたパラメータに基づいてアクセス処理性能推定関数生成部250により生成された各キャッシュ装置130のアクセス処理性能推定関数を取得することにより、分割後における各キャッシュ装置130のアクセス処理性能推定関数を生成する。

【0108】

グループ分割部950は、分割時間関数生成部940により生成されたアクセス処理性能推定関数によるアクセス処理性能の推定値の合計が、分割前の一のキャッシュ装置130のアクセス処理性能の推定値より高い場合に、一のキャッシュ装置130におけるそれぞれのグループを、複数のキャッシュ装置130における対応するグループに分割する。すなわち、グループ分割部950は、パラメータ変更部255を介してグループの分割をキャッシュ装置130に指示する。

【0109】

10

20

30

40

50

以上に示したグループ統合部 930 及びグループ分割部 950 によれば、複数のグループを 1 つのグループに統合し、又は、1 つのグループを複数に分割することにより、キャッシュ装置 130 の数をより好適に変化させることができる。更に、グループ統合部 930 及びグループ分割部 950 は、次のようにして協調動作することにより、複数のキャッシュ装置 130 の割り当てを初期状態に戻して割り当て直すことができる。

まず、グループ統合部 930 は、複数のキャッシュ装置 130 におけるそれぞれのグループを、一のキャッシュ装置 130 における対応するグループに統合する。次に、グループ分割部 950 は、一のキャッシュ装置 130 に統合されたグループを複数のキャッシュ装置 130 に分割した場合のアクセス処理性能の推定値を、異なる数のキャッシュ装置 130 に分割する場合のそれぞれについて求める。そして、グループ分割部 950 は、一の

10

#### 【0110】

以上に示したキャッシュ最適化装置 140 によれば、複数のキャッシュ装置 130 を用いることができる場合において、グループの統合・分割により最適な数のキャッシュ装置 130 を用いてアクセス処理を行わせることができる。これによりキャッシュ最適化装置 140 は、アクセス要求の到着頻度に基づいてアプリケーションサーバを稼働させる J V M の数を動的に変更し、S L A (Service Level Agreement) を保証することができる。

#### 【0111】

図 10 は、本実施形態に係るコンピュータ 900 のハードウェア構成の一例を示す。本実施形態に係るコンピュータ 900 は、ホスト・コントローラ 1082 により相互に接続される CPU 1000、RAM 1020、グラフィック・コントローラ 1075、及び表示装置 1080 を有する CPU 周辺部と、入出力コントローラ 1084 によりホスト・コントローラ 1082 に接続される通信インターフェイス 1030、ハードディスクドライブ 1040、及び CD-ROM ドライブ 1060 を有する入出力部と、入出力コントローラ 1084 に接続される ROM 1010、フレキシブルディスク・ドライブ 1050、及び入出力チップ 1070 を有するレガシー入出力部とを備える。

20

#### 【0112】

ホスト・コントローラ 1082 は、RAM 1020 と、高い転送レートで RAM 1020 をアクセスする CPU 1000 及びグラフィック・コントローラ 1075 とを接続する

30

#### 【0113】

入出力コントローラ 1084 は、ホスト・コントローラ 1082 と、比較的高速な入出力装置である通信インターフェイス 1030、ハードディスクドライブ 1040、CD-ROM ドライブ 1060 を接続する。通信インターフェイス 1030 は、ネットワークを介して他の装置と通信する。ハードディスクドライブ 1040 は、コンピュータ 900 内の CPU 1000 が使用するプログラム及びデータを格納する。CD-ROM ドライブ 1060 は、CD-ROM 1095 からプログラム又はデータを読み取り、RAM 1020 を介してハードディスクドライブ 1040 に提供する。

40

#### 【0114】

また、入出力コントローラ 1084 には、ROM 1010 と、フレキシブルディスク・ドライブ 1050、及び入出力チップ 1070 の比較的低速な入出力装置とが接続される。ROM 1010 は、コンピュータ 900 が起動時に実行するブート・プログラムや、コンピュータ 900 のハードウェアに依存するプログラム等を格納する。フレキシブルディスク・ドライブ 1050 は、フレキシブルディスク 1090 からプログラム又はデータを

50

読み取り、RAM 1020を介してハードディスクドライブ1040に提供する。入出力チップ1070は、フレキシブルディスク・ドライブ1050や、例えばパラレル・ポート、シリアル・ポート、キーボード・ポート、マウス・ポート等を介して各種の入出力装置を接続する。

【0115】

RAM 1020を介してハードディスクドライブ1040に提供されるプログラムは、フレキシブルディスク1090、CD-ROM 1095、又はICカード等の記録媒体に格納されて利用者によって提供される。プログラムは、記録媒体から読み出され、RAM 1020を介してコンピュータ900内のハードディスクドライブ1040にインストールされ、CPU 1000において実行される。

10

【0116】

コンピュータ900にインストールされ、コンピュータ900をキャッシュ最適化装置140として機能させるプログラムは、パラメータ設定モジュールと、グループ対応統計情報取得モジュールと、アクセス要求到着頻度取得モジュールと、無効化要求到着頻度取得モジュールと、アクセス要求到着確率密度関数生成モジュールと、無効化要求到着確率密度関数生成モジュールと、次アクセスヒット時間分布関数生成モジュールと、キャッシュ有効時間分布関数生成モジュールと、キャッシュヒット率推定関数生成モジュールと、アクセス処理性能推定関数生成モジュールと、パラメータ変更モジュールと、統合時間関数生成モジュールと、グループ統合モジュールと、分割時間関数生成モジュールと、グループ分割モジュールとを備える。これらのプログラム又はモジュールは、CPU 1000等に働きかけて、コンピュータ900を、パラメータ設定部200と、グループ対応統計情報取得部210と、アクセス要求到着頻度取得部215と、無効化要求到着頻度取得部220と、アクセス要求到着確率密度関数生成部225と、無効化要求到着確率密度関数生成部230と、次アクセスヒット時間分布関数生成部235と、キャッシュ有効時間分布関数生成部240と、キャッシュヒット率推定関数生成部245と、アクセス処理性能推定関数生成部250と、パラメータ変更部255と、統合時間関数生成部920と、グループ統合部930と、分割時間関数生成部940と、グループ分割部950としてそれぞれ機能させる。

20

【0117】

グループ対応統計情報取得モジュールは、グループ対応キャッシュヒット率取得モジュールと、グループ対応エントリ数取得モジュールと、アクセス対象で多数算出モジュールと、グループ対応無効化数取得モジュールとを有する。これらのプログラム又はモジュールは、CPU 1000等に働きかけて、コンピュータ900を、グループ対応キャッシュヒット率取得部300と、グループ対応エントリ数取得部310と、アクセス対象データ数算出部320と、グループ対応無効化数取得部330としてそれぞれ機能させる。

30

【0118】

キャッシュヒット率推定関数生成モジュールは、エントリ数期待値関数生成モジュール、キャッシュ充填時間算出モジュール、及び次アクセスヒット率推定関数生成モジュール、又は、エントリ数期待値関数生成モジュール、シフト周期算出モジュール、リプレース時間算出モジュール、次アクセスヒット率推定関数生成モジュール、及び次アクセスヒット率平均化モジュールを有する。これらのプログラム又はモジュールは、CPU 1000等に働きかけて、コンピュータ900を、エントリ数期待値関数生成部500、キャッシュ充填時間算出部510、次アクセスヒット率推定関数生成部520、エントリ数期待値関数生成部800、シフト周期算出部810、リプレース時間算出部820、次アクセスヒット率推定関数生成部830、及び次アクセスヒット率平均化部840としてそれぞれ機能させる。

40

【0119】

以上に示したプログラム又はモジュールは、外部の記憶媒体に格納されてもよい。記憶媒体としては、フレキシブルディスク1090、CD-ROM 1095の他に、DVDやCD等の光学記録媒体、MO等の光磁気記録媒体、テープ媒体、ICカード等の半導体メ

50

メモリ等を用いることができる。また、専用通信ネットワークやインターネットに接続されたサーバシステムに設けたハードディスク又はRAM等の記憶装置を記録媒体として使用し、ネットワークを介してプログラムをコンピュータ900に提供してもよい。

【0120】

以上に示したキャッシュ最適化装置140によれば、人手による最適設定が困難であったWebアプリケーションに関するキャッシュの構成及びポリシーの設定を効率良く行うことができる。また、本実施形態に係るキャッシュ最適化装置140によれば、LRUキャッシュ又は優先度付LRUキャッシュを備える各種のキャッシュ装置130について、キャッシュヒット率の近似値を算出することができ、算出したキャッシュヒット率に基づいて最適な設定パラメータを効率良く設定することができる。

10

【0121】

以下に、本実施形態において示した式(19)又は式(24)によるキャッシュヒット率の推定値と、イベント駆動型シミュレーションにより求めたキャッシュヒット率との比較結果を示す。この比較においては、250個のアクセス対象データをそれぞれ有する4つのグループからなるキャッシュモデルを用いて、キャッシュサイズKを100から1000まで100単位で変化させながら、アクセス要求のみ受ける場合、アクセス要求及び無効化要求を受ける場合、更にタイムアウトにより無効化される場合、及び、更に優先度を考慮する場合のそれぞれについて評価を行った。この結果、式(19)又は式(24)によるキャッシュヒット率の推定値とイベント駆動型シミュレーションによるキャッシュヒット率の差は最大で0.016であった。このことから、本実施形態に係るキャッシュ最適化装置140は、キャッシュヒット率を、イベント駆動型シミュレーションと同等の精度で関数計算により高速に求めることができると言える。

20

【0122】

以上、本発明を実施の形態を用いて説明したが、本発明の技術的範囲は上記実施の形態に記載の範囲には限定されない。上記実施の形態に、多様な変更または改良を加えることが可能であることが当業者に明らかである。その様な変更または改良を加えた形態も本発明の技術的範囲に含まれ得ることが、特許請求の範囲の記載から明らかである。

【0123】

例えば、キャッシュ装置130が、アクセス対象データの更新時にキャッシュを更新する方式を採用する場合、式(14)に示したアクセス処理性能推定関数に代えて、式(25)に示したアクセス処理性能推定関数を用いてもよい。

30

【数25】

$$C = \sum_{i=1}^M v_i \lambda_i (h_i C_i^r + (1-h_i) C_i^c) + v_i \mu_i C_i^u \quad (25)$$

【0124】

式(25)においては、アクセス対象データ更新の平均頻度を $\mu_i$ 、更新処理のコストを $C_i^u$ とし、キャッシュヒット率 $h_i$ はアクセス要求及び更新要求の合計値とする。

【図面の簡単な説明】

40

【0125】

【図1】本発明の実施形態に係る情報システム10の構成を示す。

【図2】本発明の実施形態に係るキャッシュ最適化装置140の構成を示す。

【図3】本発明の実施形態に係るグループ対応統計情報取得部210の構成を示す。

【図4】本発明の実施形態に係るキャッシュ最適化装置140の動作フローを示す。

【図5】本発明の実施形態に係るキャッシュヒット率推定関数生成部245の構成を示す。

。

【図6】本発明の実施形態の第1変形例に係るキャッシュ最適化装置140によるアクセス処理性能最適化方法を示す。図6(a)はアクセス対象データ600の配置変更によるアクセス処理性能最適化を、図6(b)はキャッシュ分割領域610のサイズ変更による

50

アクセス処理性能最適化を示す。

【図 7】本発明の実施形態の第 2 変形例に係るキャッシュ装置 130 の構成を示す。

【図 8】本発明の実施形態の第 2 変形例に係るキャッシュヒット率推定関数生成部 245 の構成を示す。

【図 9】本発明の実施形態に係る第 3 変形例に係るキャッシュ最適化装置 140 の構成を示す。

【図 10】本発明の実施形態に係るコンピュータ 900 のハードウェア構成の一例を示す。

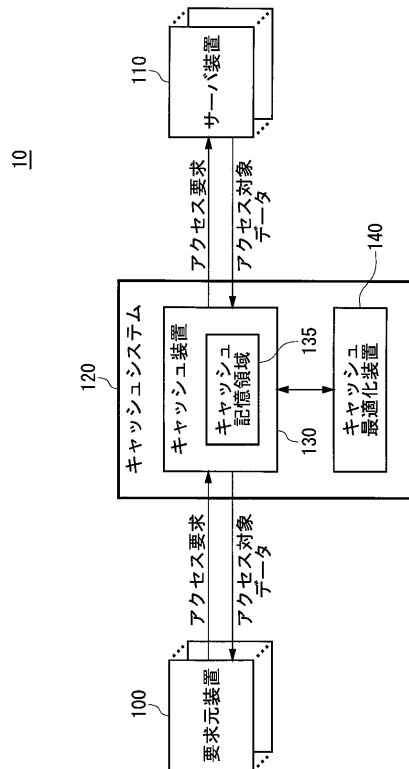
【符号の説明】

【0126】

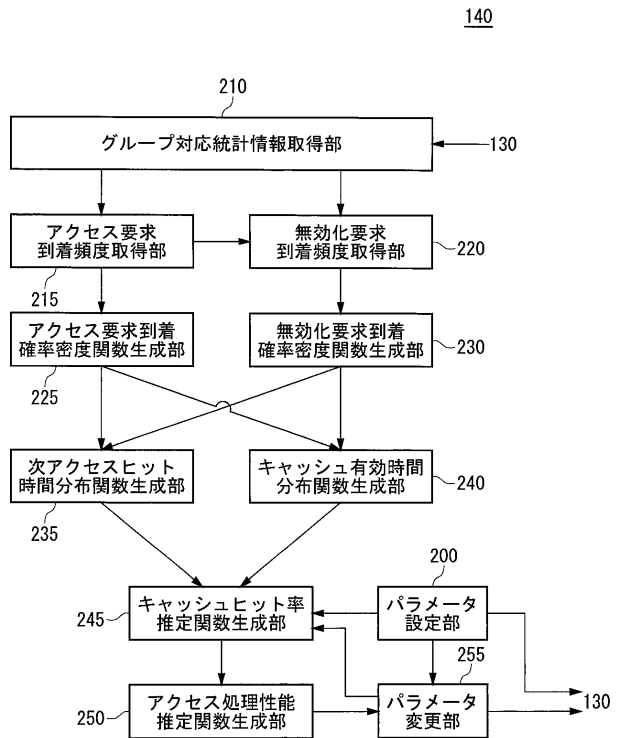
10	情報システム	
100	要求元装置	
110	サーバ装置	
120	キャッシュシステム	
130	キャッシュ装置	
135	キャッシュ記憶領域	
140	キャッシュ最適化装置	
200	パラメータ設定部	
210	グループ対応統計情報取得部	
215	アクセス要求到着頻度取得部	20
220	無効化要求到着頻度取得部	
225	アクセス要求到着確率密度関数生成部	
230	無効化要求到着確率密度関数生成部	
235	次アクセスヒット時間分布関数生成部	
240	キャッシュ有効時間分布関数生成部	
245	キャッシュヒット率推定関数生成部	
250	アクセス処理性能推定関数生成部	
255	パラメータ変更部	
300	グループ対応キャッシュヒット率取得部	
310	グループ対応エントリ数取得部	30
320	アクセス対象データ数算出部	
330	グループ対応無効化数取得部	
500	エントリ数期待値関数生成部	
510	キャッシュ充填時間算出部	
520	次アクセスヒット率推定関数生成部	
600	アクセス対象データ	
610	キャッシュ分割領域	
700	アクセス要求処理部	
710	L R U リスト管理部	
715	L R U リスト	40
720	ポインタ	
730	エントリ移動部	
740	L R U シフト部	
750	統計情報計測部	
800	エントリ数期待値関数生成部	
810	シフト周期算出部	
820	リプレース時間算出部	
830	次アクセスヒット率推定関数生成部	
840	次アクセスヒット率平均化部	
920	統合時間関数生成部	50

- 9 3 0 グループ統合部
- 9 4 0 分割時間関数生成部
- 9 5 0 グループ分割部
- 9 0 0 コンピュータ
- 1 0 0 0 CPU
- 1 0 1 0 ROM
- 1 0 2 0 RAM
- 1 0 3 0 通信インターフェイス
- 1 0 4 0 ハードディスクドライブ
- 1 0 5 0 フレキシブルディスク・ドライブ
- 1 0 6 0 CD-ROMドライブ
- 1 0 7 0 入出力チップ
- 1 0 7 5 グラフィック・コントローラ
- 1 0 8 0 表示装置
- 1 0 8 2 ホスト・コントローラ
- 1 0 8 4 入出力コントローラ
- 1 0 9 0 フレキシブルディスク
- 1 0 9 5 CD-ROM

【 図 1 】

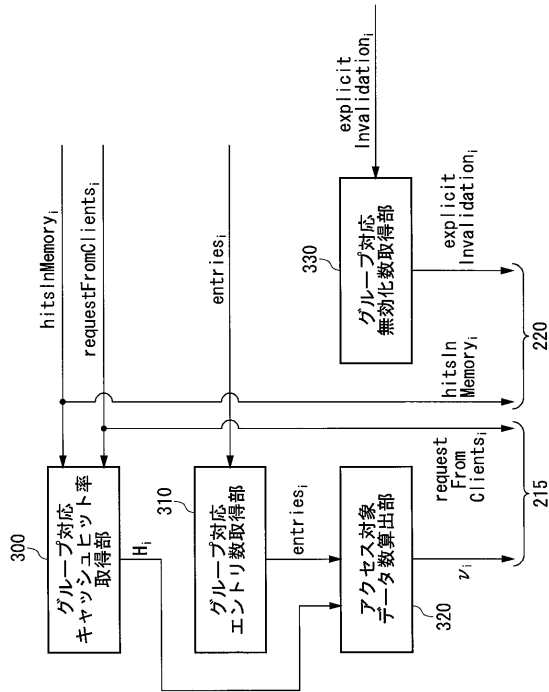


【 図 2 】

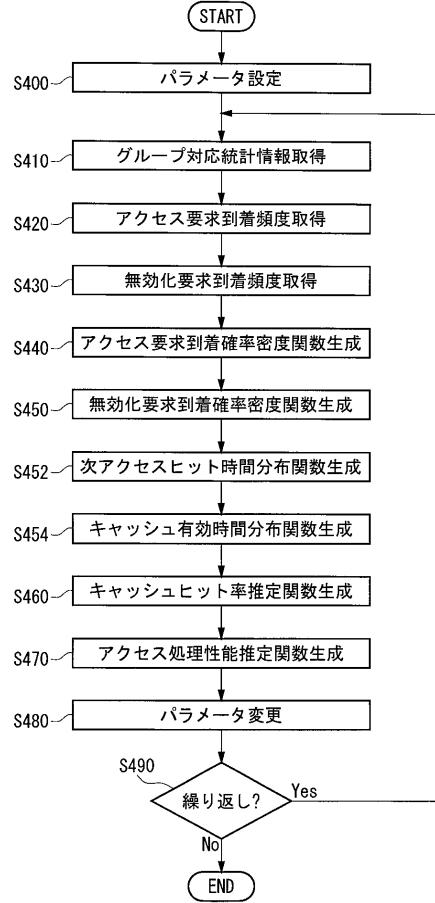


【図3】

210

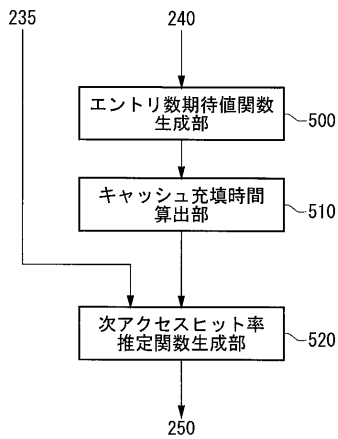


【図4】



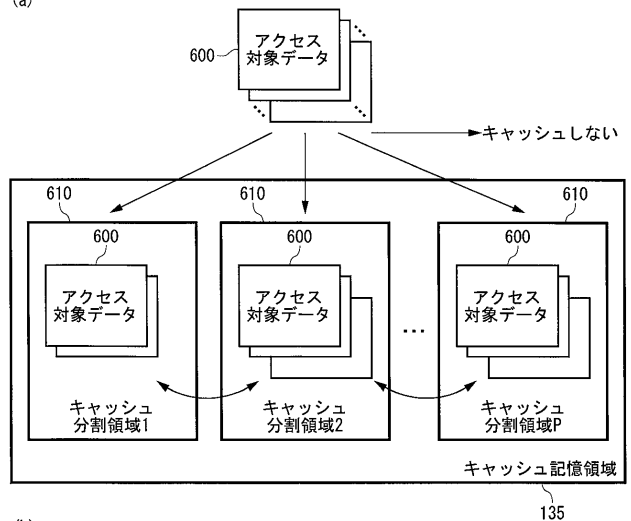
【図5】

245

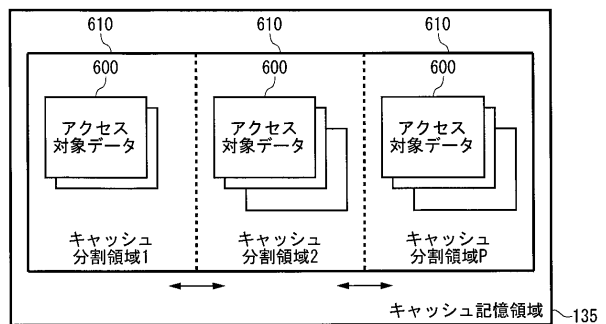


【図6】

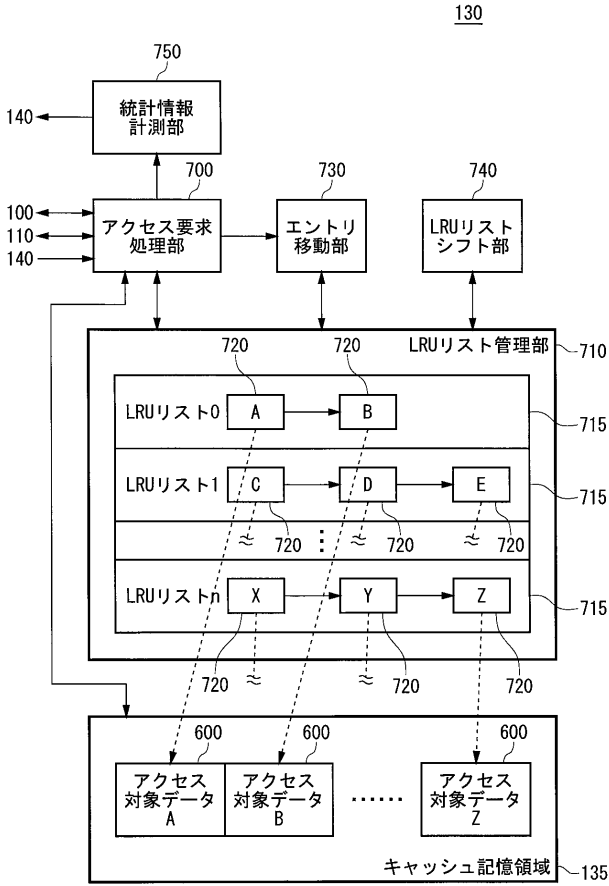
(a)



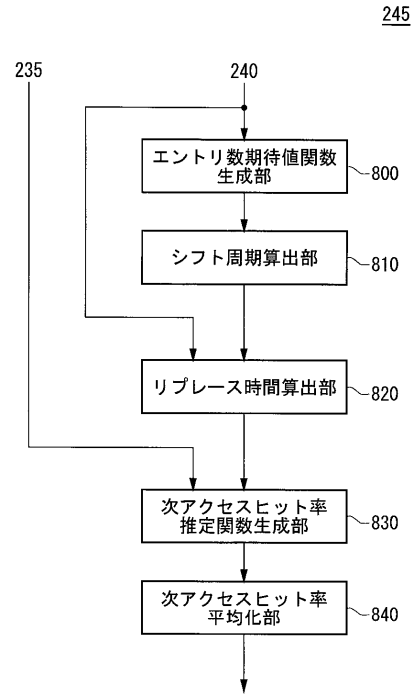
(b)



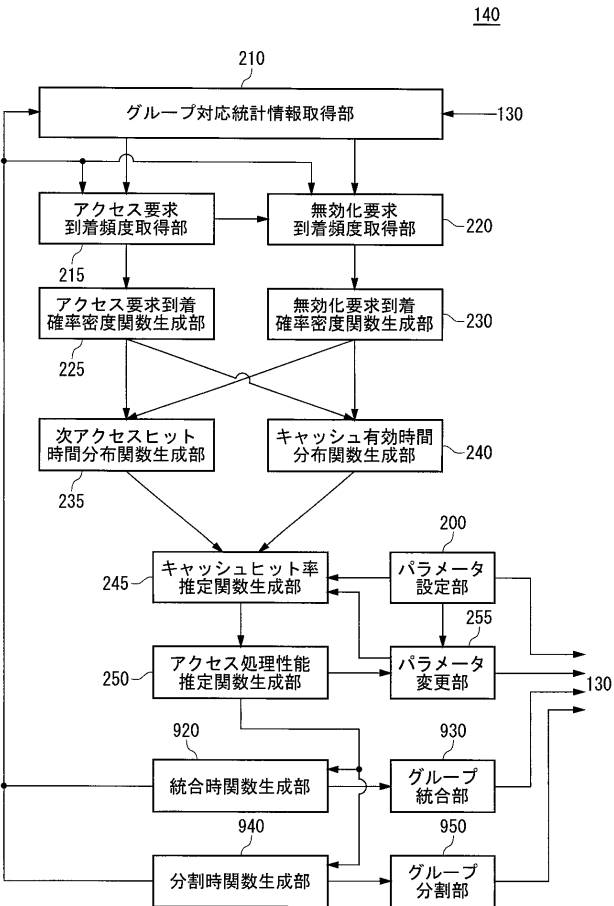
【 図 7 】



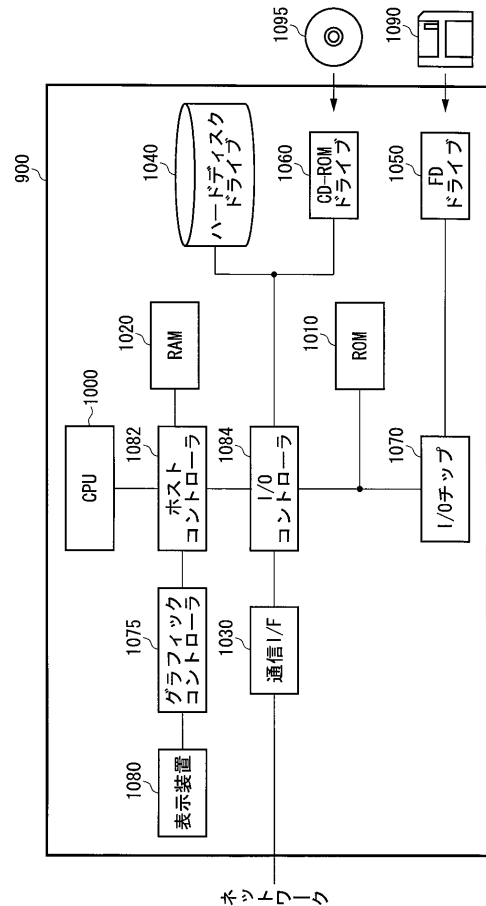
【 図 8 】



【 図 9 】



【 図 10 】





---

フロントページの続き

(74)復代理人 100104156

弁理士 龍華 明裕

(72)発明者 濱 利行

神奈川県大和市下鶴間 1 6 2 3 番地 1 4 日本アイ・ビー・エム株式会社 東京基礎研究所内

(72)発明者 平出 涼

神奈川県大和市下鶴間 1 6 2 3 番地 1 4 日本アイ・ビー・エム株式会社 東京基礎研究所内

Fターム(参考) 5B005 JJ13 KK02 KK12 LL11 MM03 QQ02 QQ04 VV04

5B065 CE21 CH20