

(19) 日本国特許庁 (JP)

(12) 特 許 公 報 (B2)

(11) 特許番号

特許第6162240号  
(P6162240)

(45) 発行日 平成29年7月12日 (2017.7.12)

(24) 登録日 平成29年6月23日 (2017.6.23)

(51) Int.Cl.

F I

G O 6 F 17/30 (2006.01)

G O 6 F 12/00 (2006.01)

G O 6 F 17/30 1 1 0 C

G O 6 F 17/30 2 4 0 A

G O 6 F 12/00 5 1 3 D

G O 6 F 12/00 5 1 4 M

請求項の数 15 (全 28 頁)

(21) 出願番号 特願2015-526899 (P2015-526899)  
 (86) (22) 出願日 平成25年8月9日 (2013.8.9)  
 (65) 公表番号 特表2015-531129 (P2015-531129A)  
 (43) 公表日 平成27年10月29日 (2015.10.29)  
 (86) 国際出願番号 PCT/EP2013/002390  
 (87) 国際公開番号 W02014/026753  
 (87) 国際公開日 平成26年2月20日 (2014.2.20)  
 審査請求日 平成28年5月9日 (2016.5.9)  
 (31) 優先権主張番号 12368020.9  
 (32) 優先日 平成24年8月14日 (2012.8.14)  
 (33) 優先権主張国 欧州特許庁 (EP)  
 (31) 優先権主張番号 13/585,286  
 (32) 優先日 平成24年8月14日 (2012.8.14)  
 (33) 優先権主張国 米国 (US)

(73) 特許権者 509228994  
 アマデウス エス. アー. エス.  
 AMADEUS S. A. S.  
 フランス共和国, エフ-06410 ビオ  
 , ソフィア アンティポリス, ルート デ  
 ユパン モンタール 485  
 485 Route du Pin Mo  
 ntard, Sophia Antipo  
 lis, F-06410 Biot, Fr  
 ance  
 (74) 代理人 100108453  
 弁理士 村山 靖彦  
 (74) 代理人 100110364  
 弁理士 実広 信哉

最終頁に続く

(54) 【発明の名称】 キャッシュ化データベース・クエリ結果の更新

(57) 【特許請求の範囲】

【請求項 1】

分散データベース・システム内の事前計算データベース・クエリ結果を更新する方法であって、前記分散データベース・システムは、前記事前計算データベース・クエリ結果を維持するデータ・キャッシュ・プラットフォーム、及び計算プラットフォーム内に維持されたデータに基づき前記事前計算データベース・クエリ結果を計算する前記計算プラットフォームを備える、方法であり：

- 前記データ・キャッシュ・プラットフォームによって、古くなった前記事前計算データベース・クエリ結果の確率を決定することであって、

- 前記決定は、確率モデル及び非同期リアルタイム・イベント発生に依存し、

- 前記確率モデルは、前記データ・キャッシュ・プラットフォーム内に維持する前記事前計算データベース・クエリ結果と仮定した実際のデータベース・クエリ結果との間の不一致をモデル化し、

- 前記リアルタイム・イベントは、前記事前計算データベース・クエリ結果の有効期限に関して非確定的であり、前記データ・キャッシュ・プラットフォーム内に維持する前記事前計算データベース・クエリ結果と仮定した実際のの前記データベース・クエリ結果との間の前記不一致に確率的な影響のみを及ぼし、

- 前記確率は、一般に、前記確率モデルに基づき決定され、可能性としては前記非同期リアルタイム・イベント発生時に修正される、決定すること；

- 古くなった前記事前計算データベース・クエリ結果の前記決定した確率に基づき前記

10

20

事前計算データベース・クエリ結果を更新するために、前記データ・キャッシュ・プラットフォームによって再計算指令を前記計算プラットフォームに自動的に発行することであって、古くなった確率が所与の閾値を超える前記事前計算データベース・クエリ結果を再計算するように指令する、発行すること；並びに

- 前記データ・キャッシュ・プラットフォームにおいて、前記再計算指令の結果として更新した前記事前計算データベース・クエリ結果を受信することを含む方法。

【請求項 2】

前記確率モデルは、統計履歴データに基づき前記計算プラットフォーム内に維持される前記データの揮発性をモデル化する、請求項 1 に記載の方法。

10

【請求項 3】

- 前記データ・キャッシュ・プラットフォームにおいて、到来する前記非同期リアルタイム・イベントが前記確率モデル内の見本となるかどうかを分析すること  
を更に含む、請求項 1 又は 2 に記載の方法。

【請求項 4】

到来する前記非同期リアルタイム・イベントが前記確率モデル内の見本となるかどうかを分析することは、

- 前記データ・キャッシュ・プラットフォーム内に維持される前記事前計算データベース・クエリ結果の持ち分の正確さを決定することであって、前記事前計算データベース・クエリ結果の前記持ち分は、到来する前記非同期リアルタイム・イベントによって影響を受ける可能性がある、決定すること；

20

- 前記確率モデルによって、前記決定した正確さを前記事前計算データベース・クエリ結果の前記持ち分の正確さの予測と比較すること；及び

- 前記決定した正確さが前記予測した正確さを所与の程度に下回る場合に、到来する前記非同期リアルタイム・イベントが前記確率モデル内の見本とならないことを決定すること  
を含む請求項 3 に記載の方法。

【請求項 5】

前記事前計算データベース・クエリ結果の前記持ち分の前記正確さを決定することは、サンプル再計算指令を前記計算プラットフォームに発行し、前記サンプル再計算指令の結果を前記データ・キャッシュ・プラットフォーム内に以前に維持されたそれぞれの前記事前計算データベース・クエリ結果と比較することを含む、請求項 4 に記載の方法。

30

【請求項 6】

- 前記確率モデル内の見本とならないと決定される前記リアルタイム・イベントに関して、影響を受ける可能性がある前記事前計算データベース・クエリ結果に関する再計算指令をできるだけ早く発行すること

を更に含む請求項 3 から 5 のうちいずれか一項に記載の方法。

【請求項 7】

- 前記確率モデル内の見本となると決定される前記リアルタイム・イベントに関して、前記リアルタイム・イベントを特定の時間期間にわたって蓄積し、実際に発生、蓄積した前記リアルタイム・イベントを前記確率モデル内の見本と比較し、実際に発生、蓄積した前記リアルタイム・イベントが前記確率モデル内の前記リアルタイム・イベント見本から所定の程度に外れる場合、影響を受ける可能性がある前記事前計算データベース・クエリ結果に対して再計算指令をできるだけ早く発行すること

40

を更に含む請求項 4 から 6 のうちいずれか一項に記載の方法。

【請求項 8】

古くなった前記事前計算データベース・クエリ結果の前記確率を決定し、前記再計算を発行するときの前記データ・キャッシュ・プラットフォームは、前記計算プラットフォーム内に維持される隣接データ・セットの群に対応する事前計算データベース・クエリ結果の格子を考慮する、請求項 1 から 7 のうちいずれか一項に記載の方法。

50

**【請求項 9】**

前記データ・キャッシュ・プラットフォームは、前記計算プラットフォームでの利用可能計算リソースの量に基づき前記再計算指令を発行する、請求項 1 から 8 のうちいずれか一項に記載の方法。

**【請求項 10】**

前記分散データベース・システムは、旅行予約システムであり、前記計算プラットフォームは、旅行利用可能性及び運賃に関する情報を維持し、前記データ・キャッシュ・プラットフォームは、前記旅行利用可能性情報及び前記運賃から計算された、価格付き旅行推奨を維持する、請求項 1 から 9 のうちいずれか一項に記載の方法。

**【請求項 11】**

前記リアルタイム・イベントは、フライト運賃の変更、飛行機座席利用可能性の変更、顧客の航空券要求及び/又は欠航を含む、請求項 10 に記載の方法。

**【請求項 12】**

前記分散データベース・システムは、前記計算プラットフォームに接続された少なくとも 1 つのアプリケーション・プラットフォームを含み、前記少なくとも 1 つのアプリケーション・プラットフォームは、前記事前計算データベース・クエリ結果を維持、構成し、前記少なくとも 1 つのアプリケーション・プラットフォーム内に記憶した前記データベース・クエリ結果は、前記データ・キャッシュ・プラットフォームが発行した前記再計算指令の結果、前記計算プラットフォームによってボピュレート及び/又は更新される、請求項 1 から 11 のうちいずれか一項に記載の方法。

**【請求項 13】**

計算プラットフォーム内に維持されるデータに基づき前記計算プラットフォームが計算した事前計算データベース・クエリ結果を維持するデータ・キャッシュ・プラットフォームであって、

- 古くなった前記事前計算データベース・クエリ結果の確率を決定し、
    - ( - 前記決定は、確率モデル及び非同期リアルタイム・イベント発生に依存し、
    - 前記確率モデルは、前記データ・キャッシュ・プラットフォーム内に維持する前記事前計算データベース・クエリ結果と仮定した実際のデータベース・クエリ結果との間の不一致をモデル化し、
    - 前記リアルタイム・イベントは、前記事前計算データベース・クエリ結果の有効期限に関して非確定的であり、前記データ・キャッシュ・プラットフォーム内に維持する前記事前計算データベース・クエリ結果と仮定した実際の前記データベース・クエリ結果との間の前記不一致に対し確率的な影響のみを及ぼし、
    - 前記確率は、一般に、前記確率モデルに基づき決定され、可能性としては前記非同期リアルタイム・イベント発生時に修正される) ;
    - 古くなった前記事前計算データベース・クエリ結果の前記決定した確率に基づき前記事前計算データベース・クエリ結果を更新するために、再計算指令を前記計算プラットフォームに自動的に発行し(古くなった確率が所与の閾値を超える前記事前計算データベース・クエリ結果を再計算するように命令する) ; 及び
    - 前記再計算指令の結果として更新した前記事前計算データベース・クエリ結果を受信する
- ように構成したデータ・キャッシュ・プラットフォーム。

**【請求項 14】**

- 到来する非同期リアルタイム・イベントが前記確率モデル内の見本となるかどうかを分析し ;
- 前記確率モデル内の見本とならない前記リアルタイム・イベントに関して、それぞれの特定の前記事前計算データベース・クエリ結果に関する再計算指令をできるだけ早く発行し ;
- 前記確率モデル内の見本となる前記リアルタイム・イベントに関して、前記リアルタイム・イベントを特定の時間期間にわたって蓄積し、実際に発生、蓄積した前記リアルタ

10

20

30

40

50

イム・イベントを前記確率モデル内の前記リアルタイム・イベント見本と比較し、実際に発生、蓄積した前記リアルタイム・イベントが前記確率モデル内の前記リアルタイム・イベント見本から所定の程度に外れる場合、影響を受ける可能性がある前記事前計算データベース・クエリ結果に関する再計算指令をできるだけ早く発行するように更に構成した請求項 13 に記載のデータ・キャッシュ・プラットフォーム。

【請求項 15】

コンピュータ・プログラム命令を中に記憶させた非一時的コンピュータ可読記憶媒体であって、前記コンピュータプログラム命令がコンピュータ・システム上で実行されると、前記コンピュータ・システムは：

- 古くなった事前計算データベース・クエリ結果の確率を決定し、10
  - ( - 前記決定は、確率モデル及び非同期リアルタイム・イベント発生に依存し、
  - 前記確率モデルは、前記コンピュータ・システム内に維持する前記事前計算データベース・クエリ結果と仮定した実際のデータベース・クエリ結果との間の不一致をモデル化し、

- 前記リアルタイム・イベントは、前記事前計算データベース・クエリ結果の有効期限に関して非確定的であり、前記コンピュータ・システム内に維持する前記事前計算データベース・クエリ結果と仮定した実際の前記データベース・クエリ結果との間の前記不一致に対し確率的な影響のみを及ぼし、

- 前記確率は、一般に、前記確率モデルに基づき決定され、可能性としては前記非同期リアルタイム・イベント発生時に修正される)；20

- 古くなった前記事前計算データベース・クエリ結果の前記決定した確率に基づき前記事前計算データベース・クエリ結果を更新するために、再計算指令を自動的に発行し(古くなった確率が所与の閾値を超える前記事前計算データベース・クエリ結果を再計算するように指令する)；及び

- 前記再計算指令の結果として、更新した前記事前計算データベース・クエリ結果を受信する、非一時的コンピュータ可読記憶媒体。

【発明の詳細な説明】

【技術分野】

【0001】

本発明は、データベース技術の分野を対象とする。より詳細には、本発明は、データベース・クエリ結果を事前計算し、キャッシュ化すること、及びこうした結果を最新に保持する方策に関する。30

【背景技術】

【0002】

データベース技術における一般的な問題は、データベース・クエリに対する短い応答時間を確実にすることであり、このことは大量のデータ処理を要する。例えば、電力を消費するそのような計算処理は、いわゆる「オープン・クエリ」に回答して実施する必要がある、この「オープン・クエリ」には、ほんのわずかな入力情報しか含まず(例えば 1 ダースの可能なパラメータのうち 1 つ又は 2 つのパラメータが特定される、及び/又は特定されたパラメータの値域が広い)、したがって、概して多数の結果をもたらす。ハードウェアの性能を増大させることによってデータ処理を高速化する可能性には、限度がある。したがって、大量データ処理の基礎をなす機構を改善することが注目を引き付けている。40

【0003】

クエリ時間を短縮させる 1 つの一般的なアプローチは、予期されるクエリを事前計算し、対応するクエリ結果をキャッシュ・システム内に維持することである。すると、クエリは、実際は大規模データに基づいて処理されず、キャッシュ・システムに誘導される。

【0004】

しかし、そのようなキャッシュ化アプローチに伴う別の問題は、キャッシュ化結果によって応答されるクエリが、対応する大規模データに基づく状態を正確に反映することを確実にするために、事前計算クエリ結果を最新に保持することである。基礎をなすデータを50

変更した場合、キャッシュ化クエリ結果は、古くなり、キャッシュ・システムは不正確な結果を返信する。したがって、キャッシュ・システムをいかに最新に保持できるかという方策が必要とされる。

【 0 0 0 5 】

例えば全データ・ドメインを頻繁に再計算すること、再計算スケジュールを手動で確率、維持すること、及びデータが古くなりすぎたときにデータを再計算すること等、様々な比較的単純な更新方策が従来技術において公知である。

【 0 0 0 6 】

例えばW O 0 1 / 3 3 4 7 2 及びW O 0 2 / 2 5 5 5 7 に記載されるように、いくぶんより高度な更新方策が開発されている。

10

【 0 0 0 7 】

W O 0 1 / 3 3 4 7 2 は、旅行計画システム内で使用される利用可能性システムに関する。システムは、航空会社の座席に関する利用可能性情報の入力を含むキャッシュを含む。キャッシュ・マネージャは、キャッシュ内の情報を正確、最新、完全に保持する、或いはできるだけ有用に保持するためにキャッシュ内の入力情報を管理する。キャッシュに誘導されたクエリに回答して、キャッシュ・マネージャは、記憶した回答が古くなったかどうかを決定し、このケースであった場合、利用可能性クエリを利用可能性情報元に送信する。修正すべきキャッシュの入力は、外部システムからの非同期通知によって得られ、確定モデル、予測モデル又は統計モデルによって決定される。

20

【 0 0 0 8 】

同様に、W O 0 2 / 2 5 5 5 7 は、情報元から受信した情報を、将来の顧客の要求等将来の使用のためにキャッシュ化する情報抽出システムに関する。キャッシュをポピュレートする及び／又は現在キャッシュ化されている情報を更新するために、能動的クエリを生成できる。航空会社の情報システムでは、能動的クエリは、出発時間の近さ、キャッシュ化データの寿命、航空機の残席、休日若しくは特別なイベント、又は装備の種類等の統計値又は予測表示に基づいて指令される。更に、更新は、A V S メッセージ等の航空会社からの外部通知によって受信される。

【 0 0 0 9 】

更に、W O 9 9 / 2 2 3 1 5 は、統計に基づく確率モデルの使用によりキャッシュ内のドキュメントを自動的にリフレッシュする機構を記載している。各ドキュメントに対して、キャッシュは、キャッシュ化オブジェクト  $i$  が特定の時間  $t$  で古くなる（即ちサーバがそのオブジェクトを変更している）確率  $P_{s i}(t)$ 、及び要求時間  $h$  によってユーザがオブジェクト  $i$  を要求する確率  $P_{r i}(h)$  を決定する。キャッシュは、最も高い積  $P_i = P_{s i}(t) \times P_{r i}(h)$ 、即ち古くなったオブジェクトを次の要求を有するユーザに返信する確率、を有するこうしたオブジェクトをリフレッシュする。こうした確率値を維持するために、キャッシュは、サーバ更新EUI間の推定平均区間等のキャッシュ化オブジェクトのための履歴統計値を維持、追跡する。オブジェクトのEUIは、例えばサーバがオブジェクト自体を更新する場合に更新される、又はオブジェクトは、オブジェクトの推定平均リフレッシュ時間が経過した後は更新されない。

30

【 先行技術文献 】

40

【 特許文献 】

【 0 0 1 0 】

【 特許文献 1 】 W O 0 1 / 3 3 4 7 2

【 特許文献 2 】 W O 0 2 / 2 5 5 5 7

【 特許文献 3 】 W O 9 9 / 2 2 3 1 5

【 発明の概要 】

【 0 0 1 1 】

本発明によれば、分散データベース・システム内で事前計算データベース・クエリ結果を更新する方法が提供される。分散データベース・システムは、事前計算データベース・クエリ結果を維持するデータ・キャッシュ・プラットフォーム、及び計算プラットフォーム

50

ムであって、計算プラットフォーム内に維持されるデータに基づき事前計算データベース・クエリ結果を計算する計算プラットフォームを備える。データ・キャッシュ・プラットフォームは、古くなった事前計算データベース・クエリ結果の確率を決定する。この決定は、確率モデル及び非同期リアルタイム・イベント発生に依存する。確率モデルは、データ・キャッシュ・プラットフォーム内に維持される事前計算データベース・クエリ結果と仮定した実際のデータベース・クエリ結果との間の不一致をモデル化する。リアルタイム・イベントは、事前計算データベース・クエリ結果の有効期限に関して非確定的であり、データ・キャッシュ・プラットフォーム内に維持される事前計算データベース・クエリ結果と仮定した実際のデータベース・クエリ結果との間の不一致に対し確率的な影響のみを及ぼす。確率は、一般に、確率モデルに基づき決定され、可能性としては、非同期リアルタイム・イベント発生時に修正される。データ・キャッシュ・プラットフォームは、再計算指令を計算プラットフォームに自動的に発行し、古くなった事前計算データベース・クエリ結果の決定された確率に基づき事前計算データベース・クエリ結果を更新する。他のものよりも古くなった高い確率を有する事前計算データベース・クエリ結果は、再計算するように指令される。再計算指令の結果、データ・キャッシュ・プラットフォームは、更新された事前計算データベース・クエリ結果を受信する。

10

【0012】

別の態様によれば、データベース・キャッシュ・プラットフォームは、上記の方法を実行するように構成される。

【0013】

20

別の態様によれば、コンピュータ・システム上で実行されると上記した方法を実施するように構成した非一時的コンピュータ可読記憶媒体を提供する。

【0014】

更なる態様は、従属請求項に示される。

【図面の簡単な説明】

【0015】

本発明を添付の図面を参照しながら説明する。同様の参照番号は、一般に同一又は機能的に同様の要素を示す。

【0016】

30

【図1】分散データベース・システムの全体概要図である。

【図2】一実施形態に従った分散データベース・システムのより詳細な図である。

【図3】一実施形態に従ったキャッシュ・プラットフォームの構成要素の図である。

【図4a】確率モデルによる経時的な揮発性効果を可視化した図である。

【図4b】リアルタイム・イベントにตอบสนองした確率モデルの修正を可視化した図である。

【図5】本方法の一実施形態に従った流れ図である。

【図6】一実施形態に従った再計算のための例示的リソース利用可能性の概略図である。

【図7】一実施形態に従ったキャッシュ・プラットフォーム・コンピュータの概略図である。

【発明を実施するための形態】

【0017】

40

図面に基づく詳細な説明を参照する前に、図1に関するいくつかのより一般的な態様を最初に示す。

【0018】

基礎をなす大量のデータに基づいた計算を必要とするデータベース・クエリ又はバッチ計算要求を扱い可能にするために、予期されるクエリに対応するデータベース・クエリ結果は、一般に事前計算され、キャッシュ化される（後に、用語「クエリ」は、トランザクション・クエリ等のあらゆる種類の情報抽出要求、バッチ計算及び他の形態のための要求を含む一般用語として使用される）。キャッシュ化結果は、記憶され、実際に発生したクエリにตอบสนองして問合せ中の実体に返信される。図1は、抽象的レベルでのそのようなデータベース・システム1を示す。基本データは、キャッシュ・プラットフォーム2に接続し

50

た計算プラットフォーム3内に保持される。キャッシュ・プラットフォーム2は、再計算指令を計算プラットフォーム3に発行し、この計算プラットフォーム3は、事前計算データベース・クエリ結果を記憶させたキャッシュ・プラットフォーム2に対応する結果を返信する。

#### 【0019】

事前計算データベース・クエリ結果をキャッシュ化するこのアプローチは、基礎をなすデータ・ドメインのデータが経時的に変化し得るという一般的な問題をもたらし、したがって、キャッシュ化事前計算データベース・クエリ結果を古くする。以下、依然として最新である、即ち対応するリアルタイム計算同等物（利用可能なキャッシュ化事前計算結果を有さず、要求に応じて実際に計算されることになる結果）と一致するキャッシュ化クエリ結果を「正確な」キャッシュ化結果と呼ぶ。したがって、キャッシュがキャッシュ化クエリ結果の基礎をなすデータ・ドメインの現在の状況を正確に表す場合、キャッシュは、「一般に」正確である。

10

#### 【0020】

概して、キャッシュに基づく正確な結果を返信するために、データベース・クエリに回答して問合せ中の実体に提供されるキャッシュ化データベース・クエリ結果とそれらのリアルタイム計算同等物の間に高度の相関関係を維持することが望まれる。しかし、同時に、再計算によって生じる計算リソースの消費を最小化する、即ち、依然として正確なキャッシュ化クエリ結果の再計算等、あらゆる不必要な再計算を回避することが望ましい。計算リソースには限度があり、一般に、全キャッシュ化クエリ結果を常に再計算するには十分な計算リソースではない。したがって、キャッシュの正確さと利用可能な計算能力の利用との間に兼ね合いを見出す必要がある。

20

#### 【0021】

事前計算データベース・クエリ結果のキャッシュを最新に保持する単純なアプローチにはいくつかの欠点がある。

#### 【0022】

データ及び利用可能な計算リソースの量に応じて全データ・ドメインを頻繁に、例えば1日に1回再計算すれば、キャッシュの正確さとリアルタイム応答との間に妥当なバランスを保証する可能性がある。しかし、このアプローチは、ほとんど拡張可能でなく、ハードウェアのリソース消費の点で非効率的である。具体的には、基礎をなす対応データが変更されていないので、依然として有効であるこれらのクエリ結果も再計算される。

30

#### 【0023】

人間の管理者が手作業でどのクエリ結果をどの時間で再計算すべきかを決定する再計算スケジュールを作れば、特定の目的に対して効率的であることを証明できるが、このことは厳密であり、柔軟性がない。スケジュールの基礎をなす仮定及び条件が変更された場合、スケジュールを再度作る必要がある。スケジュールは、基礎をなすデータ・ドメイン内に大きな変化がある場合に起こり得るキャッシュの質の急激な低下を動的に追跡することもできない。更に、スケジュールは、例えば客観的な品質基準が失われるためにそのようなスケジュールを手作業で設計するのが困難であり、人件費の点で維持が困難である。

#### 【0024】

また別のアプローチは、データが古くなりすぎたときにデータを再計算することである。しかし、基礎をなすデータ及び事前計算すべきクエリ結果の性質によっては、「古さ」の良好な閾値を評価することが困難である場合がある。

40

#### 【0025】

再計算をより効率的にするために、メトリクスは、再計算がどのくらい「不必要」であるかを評価するように定義されるべきである。例えば、計算クエリ結果の半分未満が古くなっていることが判明した場合、大量の事前計算全体を毎日繰り返し行う価値はない。一方、特定のクラスのクエリ結果が頻繁に変化することがわかっている場合、この特定のクラスのクエリ結果を1日に数回再計算すれば、正確さに関して有益であることがある。したがって、正確さに関連する利益及び再計算の費用の両方を考慮して、クエリ結果の正し

50

さを判断又は推定する有効な方法が必要である。

【0026】

本明細書で提示するキャッシュ更新方策によれば、データベース・クエリ結果の再計算は、キャッシュ化データベース・クエリが古くなる確率に基づき判定され、即ち別の再計算によって得られる結果とは異なる可能性がある。不正確さに対し少なくとも一定の、予め決定された確率を有するキャッシュ化クエリ結果のみが再計算される一方で、おそらくは依然として基礎をなすデータを正確に反映する他のキャッシュ化クエリ結果（即ちこれらは古くなる確率がより低い）は、再計算されない。

【0027】

本明細書で提示するキャッシュ更新の方策は、第1の態様として、予測モデルに基づき事前計算データベース・クエリ結果のキャッシュ全体の正確さを推定する手段に依拠する。第2の態様として、事前計算データベース・クエリ結果は、モデルに基づく再計算方策が実際のリアルタイム（及びリアルライフ）イベント発生時に依然として有効であることを検証することにより、これらの推定が一般に現実のものとは一致するかどうか確認もされ、このリアルタイム（及びリアルライフ）イベントは、例えばキャッシュ化クエリの基礎をなすデータのかなりの部分が変化している、及び - こうした変化のために - 対応するキャッシュ化クエリも古くなっていることの指摘として働くことができる。

【0028】

キャッシュの正確さの推定が一般に依拠する予測モデルは、キャッシュ化クエリ結果と仮定した実際のデータベース・クエリ結果との間の不一致をモデル化し、即ち、予測モデルは、あらゆるキャッシュ化クエリ結果の正確さ又は不正確さを概算する。モデルは、例えばキャッシュ化結果の経時的に起こりそうな揮発性をモデル化する。キャッシュ化結果の揮発性に対する仮定は、それぞれのデータ・ドメインの主題に対する（過去の）現実世界の経験から結論を下し、推断する。したがって、予測モデルは、一般に、事前計算データベース・クエリ結果の揮発性及び／又は正確さの点で現実のものの見本である。

【0029】

様々な種類のモデルを採用できる。例えば、予測モデルは、科学モデリングとして公知の構造化モデル化方法によって生成してある概念モデルであってもよい。モデル化には、現実世界環境の態様及びイベント（ここでは、例えば：事前計算キャッシュ化クエリ結果の揮発性、及び経時的に低下する正確さに影響を及ぼす態様及びイベント）の特定、選択を必要とし、モデルの目的に特に関連がないことがある他の態様は除いたままにする。更に、特定、選択された態様は、抽象化され、コンピュータ・プログラム及び／又は数式等の処理可能な形態で定式化／構築できる。そのような概念モデルは、事前計算データベース・クエリ結果の経時的な挙動（即ち揮発性／正確さ）をシミュレートするシミュレーションの手段として働くこともできる。代替的に、採用する予測モデルは、例えば過去の所与の時間期間（例えば3ヶ月）における事前計算クエリ結果の揮発性／正確さの監視及び評価に基づく履歴統計モデルであってもよく、決定した傾向を推断して将来にあてはめる。そのような履歴統計モデルの特定の例を、以下で更に示す。

【0030】

例えば、基礎をなすデータは、飛行機旅行ドメインに位置し、出発及び目的地空港、航空会社、出発日及び帰還日、運賃、予約クラス等、フライトに関する情報を含むことができる。この飛行機旅行に関連するデータは、計算プラットフォームに保持され、飛行機のフライトの利用可能性及び価格に対する知識を得るために顧客が問い合わせる。基本フライト・データに基づく価格の計算は、リソース及び時間を消費するものである。したがって、実際の価格は、事前計算され、本発明のキャッシュ・プラットフォーム内でキャッシュ化される。この例では、確率モデルは、フライト価格の経時的な揮発性をモデル化する。

【0031】

そのようなモデルを構築するのに必要な知識は、出発日前のフライト価格の挙動及び展開に関する現実世界の経験から入手できる。例えば、フライト価格は、それぞれの出発日

10

20

30

40

50



前の１ヶ月前の期間にわたって比較的安定したままであるが、出発日前のその月の間はより揮発性になることを知ることがある。したがって、確率モデルは、来月に到来するフライトに属する事前計算キャッシュ化価格が、更に遠い将来のフライトに関連する事前計算価格よりも頻繁に再計算すべきであることを示す。

#### 【 0 0 3 2 】

確率モデルを使用してキャッシュの正確さをモデル化することに加えて、リアルタイム・イベントに反応性であることによりキャッシュの正確さの急激な降下を防止する。再計算判定は、キャッシュ化クエリ結果の正確さに影響を与え得る事前計算リアルタイム・イベントの受信時に改良される。リアルタイム・イベントは、非同期式である、即ちリアルタイム・イベント発生時点が予め決定されない - リアルタイム・イベントは、常に発生できる。到来するリアルタイム・イベントを受信、処理可能にするために、キャッシュ・データ・プラットフォーム 2 は、通信ソースへの外部インターフェースを備え、この外部インターフェースは、それに応じてキャッシュ・データ・プラットフォームに関連情報を通知する。

10

#### 【 0 0 3 3 】

そのようなリアルタイム・イベントは、予測モデル内で考慮されない特定の状況に関連できる。例えば、キャッシュ化価格の一部分は、販売促進によって影響を受ける場合がある一方で、他の価格は、１年のうちの特定時間（ホリデー・シーズン、クリスマス等）でより揮発性になることがある。展示会、スポーツ・イベント等のような「例外的」状況、ストライキ又は自然災害等の無作為なイベントも、「通常の」モデル因果律の基礎をなす仮定を変更し得る。これらの特定の影響は、そのような例外的状況の見本となるそれぞれのリアルタイム・イベントに応答してキャッシュ化クエリ結果が古くなる確率を決定する際に考慮できる。代替的に、展示会、ホリデー・シーズン、スポーツ・イベント等の予定されたイベントの影響をそのイベント日の前の程よい時に確率モデルに導入できる。

20

#### 【 0 0 3 4 】

本明細書で提示する更新方策は、「不確かな」イベントを考慮できることに留意するのが重要であり、即ち、そのようなイベントとは、確実さを有する１つ又は複数の事前計算キャッシュ化クエリ結果を無効にしないが、キャッシュ化データベース・クエリ結果が古くなる確率が増大する可能性があることのみを示すイベントである。言い換えれば、これらのイベントは、キャッシュ化クエリ結果の正確さに関して非確定的であり、キャッシュ・プラットフォーム 2 内に維持されるキャッシュ化クエリ結果と、仮説的再計算から得られる仮定の実際のデータベース・クエリ結果との間の不一致に対し確率的な影響のみを及ぼす。このことは、A V S メッセージが例えば特定のフライトが欠航になっていることを示す W O 0 1 / 3 3 4 7 2 及び W O 0 2 / 2 5 5 5 7 に記載の提案とは異なる。したがって、そのような A V S メッセージの受信時に、それぞれの飛行機の座席がもはや利用不可能であることが確実にわかる。

30

#### 【 0 0 3 5 】

例えば、上述の旅行関連データ・ストレージのシナリオを参照すると、キャッシュ化クエリ結果の正確さに影響を与える可能性のあるリアルタイム・イベントは、運賃更新とすることができる。運賃は、出発及び目的地都市、予約クラス、フライトの種類（片道又は往復）、金額、及び実際に適用される運賃に対して満たさなければならない制約を定義する規則等のパラメータを含むデータ・セットである。したがって、運賃は、特定フライトに対する価格計算のための基本データの見本となる。特定の起点 - 目的地 - 都市ペアの運賃が航空会社によって更新された場合、この都市ペアに関する事前計算キャッシュ化フライト価格はもはや正しくない可能性が増大する場合がある。しかし、データ・キャッシュ・プラットフォーム 2 の観点からは、このことは確実ではない。というのは、事前計算時に事前計算プラットフォーム 3 が実際に適用した運賃は、データ・キャッシュ・プラットフォーム 2 には未知であるためである。例えば、以前の事前計算に適用した運賃は、実際には変更されていない場合があり、運賃変更イベントによって示される運賃変更は、以前関係した運賃が依然として適用されるという事実を変更せず、したがって、前に計算した

40

50

価格は、依然として有効である。又は、以前適用した運賃は実際に変更されるが、- 変更により - 今度は別の運賃が問題のフライト価格の計算に適用され、このフライト価格は、結局、キャッシュ化価格が実際には依然として有効であるという効果を有する。

【 0 0 3 6 】

したがって、そのようなリアルタイム・イベントの観察時に、データ・キャッシュ・プラットフォーム 2 は、特定のキャッシュ化クエリ結果が現在古くなっているという非確定的な可能性を用いて推測できるだけであり、キャッシュ化クエリ結果を再計算してキャッシュを正確に保持するようにすることが有利と思われる。しかし、このことは確実な事実ではなく、それぞれのキャッシュ化クエリ結果は、- 古くなったそれらの確率は増大しているにもかかわらず - 実際には依然として正しいということは十分にあり得る。

10

【 0 0 3 7 】

キャッシュ化データベース・クエリ結果が古くなる確率の決定は、2つの論理ステップで実施される：一般に、第1の論理レベルでは、確率は確率予測モデルの使用により特定される。その後、第2の論理レベルでは、こうして決定された確率は到来するリアルタイム・イベントに応答して修正できる。

【 0 0 3 8 】

このように決定された確率に基づき、データ・キャッシュ・プラットフォーム 2 は、2つの実体間の適切なネットワーク・インターフェースを介して再計算プラットフォーム 3 に再計算指令を自動的に生成、発行する（図1を参照）。一般に、再計算指令は、こうしたキャッシュ化クエリ結果に関して生成され、このキャッシュ化クエリ結果は、古くなる確率がより低い他のキャッシュ化クエリ結果よりも、古くなる確率が高い。この一般的な経験則は、確率の閾値の使用により実施できる：古くなる確率がそのような閾値を超えたと決定されるキャッシュ化クエリ結果は、再計算する必要がある。したがって、それぞれの再計算指令を送信する。古くなる確率がそのような閾値にある又はそのような閾値を下回ると決定されたキャッシュ化クエリ結果は、依然として正確である可能性があると考えられ、したがって再計算する必要がない。したがって、これらのキャッシュ化クエリ結果に対して再計算指令は発行されない。

20

【 0 0 3 9 】

特定時間における利用可能な計算能力は、再計算指令を送信する前にデータ・キャッシュ・プラットフォーム 2 によって考慮に入れられる。利用可能リソースを考慮可能にするために、データ・キャッシュ・プラットフォーム 2 は、計算プラットフォーム 3 の能力利用率の度合い及び/又はスケジュール、並びに自由計算リソースの度合い及び/又はスケジュールについての知識を有する必要がある。関連情報は、2つのプラットフォーム間の通信リンクを介してポピュレートされる。

30

【 0 0 4 0 】

再計算指令の受信に応答して、再計算プラットフォーム 3 は、それぞれのクエリ結果を再計算し、このクエリ結果をデータ・キャッシュ・プラットフォーム 2 に返信し、このデータ・キャッシュ・プラットフォーム 2 では、クエリ結果は記憶され、確率の監視及び決定が繰り返される。

【 0 0 4 1 】

40

特定のリアルタイム・イベントに応答して再計算判定を修正すべきか又は無効にするかを判定する前に、確率モデルと発生するリアルタイム・イベントとの間の関係を考慮することが好ましい。基本的に、リアルタイム・イベントは、リアルタイム・イベントがすでに確率モデル内に存在するかどうか、及びリアルタイム・イベントがすでに確率モデル内にどの程度存在するかを分析すべきである。

【 0 0 4 2 】

イベントが確率モデル内に存在するか否か、及びイベントが確率モデル内にどの程度存在するかというこの決定は、一般に、使用するモデルの種類に依存する。例えば、モデルが科学モデルの生成過程の結果である概念モデル（上記で簡単に概説した）である場合、このモデルを生成した当業者は、どの態様、イベント及び仮定がモデルに関連があると考

50

慮されてそのモデルに導入され、他のどの態様、イベント及び仮定がそれほど関連がないとみなされ、したがって、モデル内に含まれないかを認識している。この場合、したがって、どの非同期リアルタイム・イベントがモデル内の見本となり、どの非同期リアルタイム・イベントがモデル内の見本とならないかが一般にわかる。

#### 【0043】

一方、モデルが、事前計算データベース・クエリ結果の揮発性、正確さ及び／又は人気度の過去の展開及び傾向に主に基づく履歴統計モデルの場合、どのイベントが過去の事前計算データベース・クエリ結果の揮発性／正確さ／人気度に対する特定の挙動をもたらしたかが最初にわからない場合がある。しかし、この場合も同様に、特定の非同期リアルタイム・イベントが確率モデル内に含まれるか否かを概算することが可能である。1つのアプローチは、例えば、（例えばこれらの見本となるサンプルを再計算し、再計算結果をキャッシュ化クエリ結果と比較することによって）事前計算データベース・クエリ結果の実際の正確さを決定し、この実際の正確さを確率モデルが予測する正確さと比較することである。実際の（サンプル）正確さが予測した正確さよりも実質的に低い、例えば所与の閾値を下回る場合、確率モデルが特定の非同期リアルタイム・イベントを認識していない、即ち、イベントはモデル内の見本とならないことが仮定できる（或いは、モデルはより低い正確さを示したことにともなる）。他方では、実際の（サンプル）正確さが、確率モデルが予測する正確さの領域にある（即ち、正確さをわずかに下回る、正確さにある又は更には正確さよりも高い）場合、非同期リアルタイム・イベントが確率モデル内に含まれていることが仮定できる（又は、イベントは、事前計算データベース・クエリ結果の揮発性又は正確さに限定的な影響しか及ぼさないという、無視できるイベントにすぎない）。このアプローチのより詳細な例を以下で更に示す。

#### 【0044】

モデル内の十分な見本となるそのようなイベントに関して、確率の修正は必要ではない。というのは、イベント発生は、確率モデルに基づきそれぞれのキャッシュ化データベース・クエリ結果の確率を決定する際にすでに考慮に入れられているからである。一方、あるリアルタイム・イベントが確率モデル内の見本とならないことがわかった場合、そのリアルタイム・イベントは、直ちに考慮され、確率が修正され（即ち増大され）、可能性としては、それぞれのキャッシュ化データベース・クエリ結果に関する再計算指令が、修正された確率のために発行される。

#### 【0045】

任意選択で、確率モデル内に存在するリアルタイム・イベントがある程度で発生したことは、傾向を査定するために蓄積される。実際に出現し、確率モデルが一般にモデル化したリアルタイム・イベントの蓄積が突発的な逸脱（burst）を示す場合、モデル確率によって考慮されるものを超えた程度が修正され、適用可能な場合、再計算指令がそれに応じて無効になる。

#### 【0046】

任意選択で、ごく少数のキャッシュ化クエリ結果を古くする可能性がある及び／又は無関係と考慮される可能性があるイベントを除外するために、イベントは、群に蓄積、分析もされる。同じくこの理由で、イベントは記録され、経時的に収集され、集合的に扱われる。このようにして、低影響のイベントに回答して多すぎる再計算指令が発生することが防止され、したがって、計算リソース費用の不均等な増加が回避される。

#### 【0047】

要約すれば、キャッシュ化データベース・クエリ結果の正確さに少なくとも予め決定した程度を上回って影響を与える可能性のあるリアルタイム・イベントを考慮すると、キャッシュ劣化に対する高い反応性を実現する。

#### 【0048】

本発明のキャッシュ更新方策は、例えばEP2521074A1の主題であるAmadeusのマッシュ・コンピューション・プラットフォーム（MCP）と共に使用できる。MCPと共存する本発明のデータ・キャッシュ・プラットフォームを採用することによ

り、強化されたサブシステムがM C P再計算のトリガに利用可能である。M C Pが生成した旅行推奨等のデータベース・クエリ結果は、更なる分析用にデータ・キャッシュ・プラットフォームに複製、記憶される。再計算判定は、確率モデルに基づき行われ、確率モデル自体は、他のA m a d e u sサービスから入手した統計データに基づき構成できる。更に、フライト運賃変更、飛行機座席の利用可能性の変更、予約クラスの無効化、顧客の航空券要求、ユーザの品質フィードバック・イベント、欠航及び/又は同様のもの等のリアルタイム・イベントが考慮される。

#### 【 0 0 4 9 】

本発明のキャッシュ更新アプローチのための1つの例示的適用例は、仮ショッピング ( p r e - s h o p p i n g ) である。旅行を実際に予約する前に、旅行業界のエンド・ユーザは、通常、現在のフライト価格を含む利用可能なフライトについて、そのフライトを実際に予約する約束を何ら有することなく把握することを望む。仮予約 ( p r e - b o o k i n g ) 情報に対する、かなり頻繁な束縛のないそのような要求は、オープンで広範なデータベース・クエリの形態を取り、クエリ時にのみ計算されると莫大な量の計算リソースを必要とすることになる。更に、顧客は、要求した情報が顧客のクエリに応じてほぼ瞬時に伝えられることを期待する。したがって、価格付き飛行機旅行推奨等、仮ショッピングのクエリ結果は、通常、事前計算され、キャッシュ化される。したがって、旅行業界の仮ショッピングは、本明細書で提案するキャッシュ更新方策のための適切な適用例を形成する。

#### 【 0 0 5 0 】

##### 詳細な説明

次に、より詳細な説明を参照すると、図2は、例示の実施形態による分散データベース・システム1の概要図を示す。次に記載する実施形態は、旅行業界のデータベースに関連する。具体的には、計算プラットフォーム3が飛行機旅行提案に関するデータを維持し、キャッシュ・データ・プラットフォーム2が飛行機旅行提案に関連する価格を記憶し、飛行機旅行提案とは、計算プラットフォーム3が計算規則、具体的にはフライト運賃及びそれらに関連する計算規則に基づき計算したものである一実施形態が、提示される。しかし、これらの実施形態は、本発明のキャッシュ更新方策をより詳細に示す目的での例に過ぎないことを留意されたい。本明細書で提示するキャッシュ更新方策は、データ及びキャッシュ化結果の構造及び/又はセマンティクスとは無関係に、あらゆる種類のデータ及びデータベース・クエリ結果に適用できる。

#### 【 0 0 5 1 】

上記のように、分散データベース・システム1の主な実体は、データ・キャッシュ・プラットフォーム2 (以下簡単にD C Pと呼ぶ) 及び計算プラットフォーム3である。図2の例では、計算プラットフォーム3は、欧州特許出願第1 1 3 0 5 5 1 8号に開示するマッシブ・コンピューション・プラットフォーム (M C P) である。D C P 2 及びM C P 3は、少なくとも1つの通信リンクを介して結合され、この少なくとも1つの通信リンクを利用してD C P 2 からM C P 3 に再計算指令を送信し、それに応答して、事前計算された価格付き旅行推奨 (以下簡単に「価格」とも呼ぶ) をM C P 3 からD C P 2 に返信する。

#### 【 0 0 5 2 】

D C P 2 は、D C P 2 がキャッシュ化価格の正確さの確率の決定に使用するデータを組み込むための更なる通信インターフェースを備える。これらのインターフェースは、例えば通信リンクを含み、この通信リンクは、確率モデルの基準を形成する統計データを組み込み、航空会社又は顧客販売促進キャンペーンによってポピュレートされた運賃変更及びフライト利用可能性の案内等の非同期リアルタイム・イベントを受信するためのものである。

#### 【 0 0 5 3 】

更に、分散データベース・システム1は、アプリケーション・プラットフォーム4を含むことができ、このアプリケーション・プラットフォーム4は、旅行会社等のエンド・ユ

ーザ又は外部顧客が問い合わせできるデータを編成、維持する。アプリケーション・プラットフォーム 4 は、M C P 3 とアプリケーション・プラットフォーム 4 との間のそれぞれの通信リンクを介して M C P 3 によってポピュレート、更新される。このポピュレーション及び更新は、D C P 2 が発行する再計算指令によってトリガされる。

#### 【 0 0 5 4 】

一般に上記に記載し、より具体的に以下に記載するように、D C P 2 から受信した再計算指令に応答して、M C P 3 は、旅行推奨の価格を再計算し、この価格を D C P 2 に返信する。しかし、同時に、M C P 3 は、再計算した価格付き旅行推奨をアプリケーション・プラットフォーム 4 に転送もし、この旅行推奨を記憶もする（図 2 の「旅行推奨集積」に示すように）。したがって、アプリケーション・プラットフォーム 4 も、D C P 2 が実施するキャッシュ更新方策に基づき、ユーザが問い合わせる事前計算価格付き旅行推奨をキャッシュ化する。したがって、本発明のキャッシュ更新方策は、例えばオープン・クエリに対する即時応答の形態でユーザに利益をもたらす適用例に活用される。そのような構成では、データ・キャッシュ・プラットフォーム 2 は、アプリケーション・プラットフォーム 4 のキャッシュの更新を制御、トリガするように構成した制御プラットフォームとして働く。したがって、データ・キャッシュ・プラットフォーム 2 に記憶したキャッシュ化データベース・クエリ結果は、任意のユーザ又は顧客によって実際にはアクセス又は問い合わせされず、キャッシュ更新方策が実施される制御データ基盤を形成するにすぎない。しかし、他の構成では、データ・キャッシュ・プラットフォーム 2 は、ユーザ又は顧客が直接問い合わせることもでき、又は - 言い換えれば -、本発明のキャッシュ更新方策は、個別の制御実体とは対照的に 1 つ又はいくつかのアプリケーション・プラットフォーム（複数可）4 に直接実装できる。

#### 【 0 0 5 5 】

アプリケーション・プラットフォーム 4 は、例えば仮ショッピング・アプリケーション・プラットフォーム、運賃分析アプリケーション・プラットフォーム及び他のプラットフォームを備える。仮ショッピング・アプリケーション・プラットフォームは、フライト利用可能性及び価格についての情報を望むエンド・ユーザが問い合わせる。例えば、エンド・ユーザは、ホリデーシーズン中の、ニース発、5 0 0 ユーロ未満の旅行提案の価格の概要を得るために、仮ショッピング・アプリケーションにクエリを指示できる。本発明のキャッシュ更新方策と一致して更新される仮ショッピング・アプリケーション内でキャッシュ化した事前計算価格付き旅行推奨のために、それぞれのフライトの価格は、クエリの発生時に計算する必要がない。そうではなく、こうしたかなり不特定の制約を満たす旅行提案のリストは、仮ショッピング・アプリケーション・プラットフォーム内でキャッシュ化した価格付き旅行推奨に基づき非常に迅速に返信できる。次に、ユーザは、ユーザに最適である返信されたリストから旅行を選択し、次にこの旅行を実際に予約するための更なる要求を発行できる。次に、第 2 の要求は、現在及び実際の価格を計算し、ユーザに拘束力のある提案を勧める予約エンジン（図示せず）によって処理される。

#### 【 0 0 5 6 】

次に、図 3 により示されるデータ・キャッシュ・プラットフォーム 2 の構造をよく見ると、データ・キャッシュ・プラットフォーム 2 は以下の 3 つのモジュールから構成される：

- 入力マネージャ 5 は、M C P 3 からの事前計算データベース・クエリ結果、非同期リアルタイム・イベント、及び統計データ等の他の情報等の入力を受信する責任を負い、確率モデルを供給、更新する。

- 分析器 6 は、どの確率モデルが更新すべき候補のキャッシュ化クエリ結果を決定するように構成されるかに基づいて確率モデルを利用する。

- 最後に、統合器 7 は、分析器 6 が決定した確率を修正し、- 必要な場合に - 観察されたリアルタイム・イベントに基づき確率モデルも修正する（観察されたリアルタイム・イベントは図 3 に示されない）。

#### 【 0 0 5 7 】

10

20

30

40

50

更に、DCP2は、キャッシュ化価格付き旅行推奨データを保持する内部データベース8を含む。この見本は、古くなる確率の評価及び再計算判定への判定に関連する価格情報の属性、例えば：都市ペア、出発日、滞在期間及び最終計算日等を保持するにすぎず、これらは全てMCP3が返信した計算の出力である。運賃等のMCP3の計算を実施するためにMCP3によって利用される他のデータは、DCP2にミラーリングされない。というのは、他のデータは、キャッシュ更新方策の実施に必要なではないためである。しかし、一方で、DCP2は、最初に仮定した正確さ（MCP3がたった今再計算した価格が予約のための計算とは異なる見込み）、揮発性（価格がその最終計算以来の予約のための計算とは異なる確率の指摘）及び人気度（フライトがどれくらい頻繁に検索、予約されるか）等、メタデータ属性を有するDCP2のデータ（MCP3が返信するデータ・セットの一部ではない）を向上させる。こうした属性の設定に必要なデータは、揮発性データベース10、最初の正確さデータベース11及び統計サーバ9等の外部データベースに保持される。メタデータ属性は、（以下でより詳細に説明するように）分析器6がキャッシュ化価格が古くなった可能性があるかどうかの確率を決定することに基づき確率モデルの見本となる（及びしたがって確率モデルの特定性に依存する）。

10

#### 【0058】

入力マネージャ5は、全ての異種の情報ソースを変換し、MCP3によって返信されるローカルの価格見本データベース8に集積するように構成される。入力マネージャ5は、モデル化価格に影響を与える可能性のあるイベント及び行動を記録する。こうしたイベントには、顧客販売促進及び顧客不一致フィードバックを含む。更に、欠航のようなフライト利用可能性イベントは、欠航したフライトに直接的に基づくキャッシュ化旅行推奨を無効にするだけでなく、欠航したフライトと同時刻に予定されている同じ都市ペアのフライト等の対応するキャッシュ化データにも影響を与え得る。次に、こうしたリアルタイム・イベントは、古くなる確率及び再計算判定を修正するために、リアルタイム・イベントを更に処理する統合器7に転送される。

20

#### 【0059】

価格付き旅行推奨のキャッシュ化に関与する情報量のために、符号化技法を採用することが有益である。この符号化技法により、DCP2でキャッシュ化された価格付きデータは、MCP3に保持した、基礎をなすデータ・ドメインに包括的にマッピングされ、一方で記憶リソースに関する費用を著しく低減する。確率の符号化は、例えばブルーム・フィルタを使用することにより実施される。そのような符号化の効果は2つある。第1に、ブルーム・フィルタは保存的である。ブルーム・フィルタは、例えば運賃変更を示すリアルタイム・イベントによって影響を及ぼされる可能性があるこうした価格を少なくとも及びいずれの場合にも確実に追跡可能にするが、ブルーム・フィルタは、影響を受けないと考慮される価格が実際に影響を受けない逆の場合は、誤ることがない。そのため、そのような運賃変更イベントによって影響を及ぼされる可能性のある価格を認識しない恐れがない。第2に、偽陽性量の表示は、ブルーム・フィルタの割り振られた大きさに厳密に依存し、そのため、必要に応じて偽陽性の発生を制限できる。

30

#### 【0060】

第2のモジュールである分析器6は、DCP2に保持される事前計算価格の正確さの劣化の確率モデルに基づき、キャッシュ化価格付き旅行推奨が古くなったかどうかの確率の第1の一般レベルの決定を実施する。分析器6は、上記で説明したように入力マネージャ5によって価格に加えられるメタデータを検査、評価する。したがって、このメタデータによって見本となる確率モデルは、揮発性データベース10が含む価格揮発性、最初の正確さデータベース11から組み込まれた価格の最初の正確さ、及び統計サーバ9からの人気度報告により返信されたフライト推奨の人気度に関するメトリクスを含む。分析器6は、静的確率情報のみに基づき（即ちあらゆるイベントを考慮せずに）、キャッシュ化価格に関する確率及び優先情報、即ちどの価格を優先情報を用いて再計算する必要があるかの指摘を統合器7に出力する。

40

#### 【0061】

50

より具体的には、例示的確率モデルは、事前計算データベース・クエリ結果  $x_i$  の以下のパラメータに基づく：

- 事前計算データベース・クエリ結果の寿命  $t_i$ ：M C P 3 によるこの事前計算データベース・クエリ結果の最終計算以来の時間。

- 発生及び過去の計算結果から導出できる事前計算クエリ結果の揮発性  $\lambda_i$ 。

- 揮発性  $\lambda_i$  を採用して、所与の時間後に未変更のままにいる事前計算クエリ結果：P (  $t$  後未変更 ) =  $e^{-\lambda_i t}$  に対する確率推定値を与えることができる。このことは、予期される正確さ

【 0 0 6 2 】

【 数 1 】

$$acc_i^t = e^{-\lambda_i t}$$

10

の確率、又はより一般には、古くなった事前計算データベース・クエリ結果の確率とも呼ばれる。経時的に減少する可能性が高いこの正確さの2つの例示的機能は、図 4 a に示される。

- 最初の正確さ：M C P 3 が生成する事前計算データベース・クエリ結果は、例えば M C P 3 自体がその計算をキャッシュ化（したがって古くなった）データに基礎を置く場合、計算時間でさえ必ずしも正確ではないことがある。このことは、M C P 3 が計算する事前計算データベース・クエリ結果と基礎をなす正確なデータが仮説的に生成している計算結果との間に更なる不一致をもたらす。この不一致は、それぞれのフィードバックが利用可能である場合に測定できる。この不一致は、例えば以前の計算から、事前計算データベース・クエリ結果  $x_i$  が M C P 3 による計算時に正確である確率  $a_i$  を有することを推測できる。このことは、所与の時間  $t$  後、事前計算データベース・クエリ結果が正確である確率が  $a_i e^{-\lambda_i t}$  であることを意味する。

20

- 次に、このモデルに従って D C P 2 に保持される全体の事前計算データベース・クエリ結果の正確さは、平均正確さ：

$$\text{包括的正確さ} = \text{平均} ( a_i e^{-\lambda_i t} )$$

として考慮できる。

- 事前計算データベース・クエリ結果の人気度  $p_i$ ：人気度  $p_i$  は、エンド・ユーザによるこの事前計算データベース・クエリ結果への平均アクセス頻度である。更に以下でも簡単に提示するように、ユーザが要求することが多い事前計算データベース・クエリ結果に対してより良好な正確さを達成することが望ましい場合がある。したがって、ユーザが見る全キャッシュの正確さは、各正確さの値が人気度によって重み付けされるという点で定義もできる。したがって、事前計算データベース・クエリ結果への正確なアクセスの割合は、正確な事前計算データベース・クエリ結果の予期される割合とは反対に：

30

【 0 0 6 3 】

【 数 2 】

$$\text{ユーザの正確さ} = \sum \frac{p_i}{p_{\text{tot}}} a_i e^{-\lambda_i t_i}$$

に含まれる。

40

【 0 0 6 4 】

正確さの確率モデルのこうしたパラメータは、例えば数日、数週間又は数ヶ月の履歴に基づき統計サーバ 9 から導出する統計履歴データにより設定される。あらゆる単一の事前計算データベース・クエリ結果は、事前計算データベース・クエリ結果の状態を予測し、したがって全キャッシュの質を予測するために、こうしたパラメータを用いてモデル化される。

【 0 0 6 5 】

優先順位を付け、次にどの価格を再計算するか判定するために、確率モデルの情報を利用するいくつかの方法がある。分析器 6 は、状況に応じて（例えば M C P 3 内に基礎をなす旅行データを所有する顧客との契約に従って、データ量に応じて、利用可能な計算リソ

50

ースに応じて、どの方法でキャッシュを最適化すべきかの異議に応じて) こうした方策又は方策の混合を適用するように構成可能である。以下の方策を適用できる：

- ・ 価格の正確さ：このことは、データ・ドメインの包括的な正確さを最大化することを目的とする。おそらく、不正確な価格を最初に再計算する。

- ・ 人気度で重み付けした価格の正確さ：不正確である可能性がある価格のうち、人気度があまりない価格よりもより人気度のある価格を高い優先度で再計算することになる。

- ・ 人気度及び寿命で重み付けした価格の正確さ：以前の方策と同様であるが、最終再計算の時間も考慮する。この方策は、特に再計算リソースが一般に再計算すべき価格の量と比較して制限される状況で、非常に揮発性である価格によって引き起こされる再計算の欠乏を防止する。

10

- ・ 人気の都市ペアをそれらの地理位置及び再計算時間に基づき調整する：この方策は、更に、1日の特定時間でどの都市ペアのフライトがより頻繁に問い合わせられるかという統計を考慮する。結果として、特定都市ペアのフライトがほとんどアクセスされないこれらの時間で頻繁な再計算が回避される(というのは、不正確なキャッシュ化データは、それぞれのクエリが実際にほとんど発生しない限り害を与えないためである)。

#### 【0066】

副次的な結果として、分析器6は、MCP3から受信し、DCP2に組み込んだ最近再計算した価格の値に基づき揮発性モデル・データベース10を更新する。分析器は、繰り返す再計算に基づきキャッシュ化価格の実際の揮発性を追跡できるので、分析器は、こうした統計情報を揮発性モデル・データベース10にフィードバックできる。揮発性モデルを更新するために、分析器6は、新たに計算した価格結果と以前受信した価格値との間の差異の数を計数する。こうした差異から、分析器6は、分析した価格のそれぞれの部分のための揮発性パラメータを更新する。

20

#### 【0067】

同様に、分析器6は、最初の正確さデータベース11を同様に更新できる。分析器6は、例えば新たな都市ペアからの価格が初めてDCP2に集積された場合、他の人気度の報告を求めることもできる。

#### 【0068】

価格の揮発性、正確さ又は人気度に対する履歴及び統計データそれぞれがない場合、分析器6は、できるだけ保存的であるデフォルト・パラメータを使用して処理を実施する。

30

#### 【0069】

次に第3のモジュールを参照すると、統合器7は、到来するリアルタイム・イベントを考慮することにより第2のレベルの確率決定を実施する。更に、統合器7は、再計算指令を生成し、再計算指令をMCP3に発行する実体である。統合器7は、分析器6の出力をその判定の基礎として必要とする。これらの出力は、データ・ドメインの全価格に対する再計算優先度の第1の推定を提供する。次に、統合器7は、様々なリアルタイム・イベント・ソースから収集した全情報をオーバーレイして、再計算優先度を修正する。これにより向上した再計算優先度がもたらされる。

#### 【0070】

任意選択で、統合器7は、例えば「全価格を少なくとも毎週一度再計算することを保証する」等、任意の顧客サービス・レベル契約を考慮し、それに応じて優先度を修正できる。統合器7は、内部価格データ見本8内の最も高い優先度を有するこうした入力を最初に選択し、再計算用に入力をマークする。統合器は、好ましくはMCP3で利用可能な計算リソースの知識を有するので、特定の時間間隔でMCP3により再計算できるものと同数のキャッシュ化価格を確保できる。次に、統合器7は、得られた再計算指令をMCP3に送信する。

40

#### 【0071】

リアルタイム・イベントからの情報は、厳密な統計モデル化の間、キャッシュ化データの正確さを改良する手段である。リアルタイム・イベントからの情報を使用して、予期されたもののみではなく実際に起こったことを追跡できる。このことは、統計モデルの予測

50



を制御し、予測が間違っている又は不適切であると判明した場合に1つ又は複数の予測を修正する手段である。リアルタイム・イベントのいくつかのクラスを本実施形態に対して想定できる。

#### 【0072】

行為者のイベントは、一般に選択的に（即ち時間ごとに）発生するが、再計算判定に劇的な影響を及ぼす場合がある。外部顧客は、キャッシュと、外部顧客が自身のプラットフォーム上で経験しているショッピングとの間の不一致に対してフィードバックを提供できる。このフィードバックは、統計モデルが予測する正確さを修正し、したがって必要な場合により迅速な再計算を強制するために使用できる。MCP3内に記憶されるデータの提供業者（例えば旅行を提供する旅行提供者）がいくつかの都市ペアに対するフライトを促進する販売促進キャンペーンを実施するとき、これらの価格がより揮発性であり、より頻繁に古くなることが仮定できる。したがって、販売促進キャンペーン中のこれらの価格の再計算頻度を増加させる必要がある場合がある。別の例として、MCP3のための整備操作が時折必要であることがあり、さもなければシステム1内で操作上の問題に遭遇するおそれがある。そのような場合、整備が完了する及び問題が元通りになるまで、再計算指令をより少なく生成する又は再計算指令を生成しないようにDCP2に命令できる。

10

#### 【0073】

利用可能イベントは、キャッシュ化フライトの正確さに対するリアルタイム状況を示す。イベントの記述に応じて、MCP3内の基礎をなすデータ・ドメインの特定価格が変更されており、したがってDCP2は、キャッシュ化した価格が無効になっていることを確実に知ることができる。しかし、影響が確実ではない他の価格も影響を及ぼされることがあり、したがってこれらの値段が古くなる確率を増大させることがある。例えば、「クラスの閉鎖」イベントは、特定のフライトに対する特定の予約クラスが満席になっていることを示す。このフライトの座席及びクラスは、もはや予約可能ではなく、したがってDCP2がキャッシュ化したそれぞれの価格は、確実に無効になっている。しかし、このことは、同じフライトの他のクラス、及び/又はこのフライトの直前又は直後に出発する他のフライトの同じクラスの座席がより揮発性になる可能性があることの指摘として考慮される可能性がある。したがって、こうした古くなる確率は、増大する可能性があり、これらの値段の再計算が有益である可能性がある。別の例として、ローコスト・キャリアがフライト占有率に応じて座席の価格を設定することに遭遇する。占有率の変更通知時に、それぞれのキャッシュ化価格を迅速に再計算でき、したがってキャッシュの正確さを改善/回復する。

20

30

#### 【0074】

運賃変更イベントの暗示は、推定が困難である。簡単に言うと、運賃は、特定フライトの価格を計算するのに使用する規則等の情報及び論理を含む。したがって、特定フライトの実際の価格を計算する場合、一組の運賃にアクセスし、どの運賃が関連し、実際に適用されるかを判定し、最後に価格を計算する。したがって、関係「フライト 運賃（複数可）」がある（しかし、この関係は、運賃が特定フライトに適用される制約が変化し得るので経時的に変化し得る）。しかし、もう一方の方向の関係「運賃 フライト」は、一般に追跡されない。即ち、どの運賃をどのフライトに適用するかは明らかではない。更に、運賃の変更は、基礎をなすデータ・ドメインから計算される膨大な量の価格に影響する可能性がある。

40

#### 【0075】

運賃イベントの影響を決定するために、MCP3とDCP2との間の通信を利用してMCP3が価格の計算に適用する運賃へのマッピングをDCP2に与えることができる。再計算指令に応答して価格を計算する場合、MCP3は、要求された価格を計算するためにアクセスした全運賃を記録する。次に、この情報は、運賃 フライトのマッピングに包括的に記憶され、MCP3によるあらゆる計算の間維持される。運賃変更イベントを受信すると、入力マネージャ5は、運賃変更イベントによって影響を受けるフライトを決定するためにこの包括的マッピングを検索し、これらのフライトを「更新済み」としてマークす

50

る。運賃変更は、上記で簡単に説明したように、旅行推奨価格の変更を必ずしも暗示しないことを留意されたい。

【 0 0 7 6 】

統合器 7 は、リアルタイム・イベントと基本確率モデルとの関係を考慮せずにキャッシュ化旅行推奨の再計算を始めるのではなく、イベントのキャッシュ化価格に対する影響の可能性を最初に評価する。そのようなイベントは、確率モデル内のイベントの見本に関して最初に分析される。

【 0 0 7 7 】

上記で概説したように、この分析の特定性は、使用する確率モデルの種類に依存する。概念モデルの場合、モデル生成過程からどのイベントが確率モデル内で考慮されたかがわかる。したがって、特定イベントがモデル内に存在するか否かを決定するために、到来するイベントとモデル生成過程からわかった確率モデルの特性とを照合する。

【 0 0 7 8 】

他方では、履歴統計モデルを使用でき、したがって、履歴統計モデルは、どのイベントがモデルの履歴統計データ基盤に誘導されているかをアプリアリにわからないことがある。この場合、特定イベントが確率モデル内の見本となるか否かを決定するために、以下のアプローチを取ることができる（完全を期すために：以下のアプローチは、いかなる特定の種類の確率モデルに限定されるものではなく、例えば概念モデルに採用もできる）。

【 0 0 7 9 】

全キャッシュ化事前計算データベース・クエリ結果によって形成されたデータ・ドメインの大きさに応じて、キャッシュ化データベース・クエリ結果は、以下「持ち分」(share)と呼ぶ、より小さな部分に細分される。事前計算データベース・クエリ結果が価格付き旅行推奨である例では、持ち分は、例えば起点 - 目的地ペアによって形成され、即ち、1つの持ち分は、同じ起点及び目的地の場所を有する全ての事前計算価格付き旅行推奨を含む。代替的に、データ・ドメインが大きさの限られたもののみである場合、細分を実施しない。言い換えれば、この場合、DCP2に保持される全事前計算データベース・クエリ結果を含むただ1つの持ち分がある。

【 0 0 8 0 】

非同期リアルタイム・イベントを検出した後、検出されたイベントのために、このイベントによって影響を受ける可能性がある、即ち、古くなる可能性が増大している事前計算データベース・クエリ結果を含む1つ又は複数の持ち分が特定される。少なくとも1つの持ち分の実際の正確さは、例えば、持ち分内の事前計算データベース・クエリ結果の特定又は無作為サンプルを再計算することにより、実際の正確さを推定して決定する。例えば、持ち分が10,000個の事前計算データベース・クエリ結果を含む場合、20個の事前計算データベース・クエリ結果のサンプルを再計算する。この再計算は、他の再計算と同様に実施される。DCP2は、再計算指令を生成し、MCP3に発行する。MCP3は、再計算を実施し、再計算の結果、即ち更新されたサンプル事前計算データベース・クエリ結果をDCP2に返信する。DCP2は、サンプル再計算の結果と、以前にキャッシュ化した対応するデータベース・クエリ結果とを比較し、例えば変化した（即ち古くなった若しくは不正確になっている）サンプル事前計算データベース・クエリ結果のパーセンテージ、及び/又は変化しない（即ち古くなっておらず、依然として正確であった）サンプル事前計算データベース・クエリ結果のパーセンテージを決定する。サンプル事前計算データベース・クエリ結果のこれらのパーセンテージは、持ち分全体の正確さの推定として持ち分全体に伝達される可能性があり、即ち、これらサンプルのパーセンテージから、持ち分全体がサンプルとして同様の（不）正確さであることを一般に結論付けできる。

【 0 0 8 1 】

次のステップでは、決定された持ち分の実際の正確さを確率モデルが予測する持ち分の正確さと比較する。この比較の結果に応じて、特定の非同期リアルタイム・イベントが確率モデル内の見本となるか否か、及び場合によっては特定の非同期リアルタイム・イベントがどの程度確率モデル内の見本となるかを結論付ける。決定した実際の正確さが、確率

10

20

30

40

50

モデルが予測する正確さを所与の程度を超える程に下回る場合、即ち、サンプル正確さが、モデルが予測した正確さよりも実質的に低い場合、そのイベントは、確率モデル内に含めるように考慮されない。一方、決定した実際の正確さが、確率モデルが予測する正確さを所与の程度よりも低く下回る場合、或いは両方の正確さが同一である、又は決定した実際の正確さが、予測された正確さよりも更に高い場合、そのイベントは、確率モデル内の見本となるように考慮される。両方の正確さの間の差異は、イベントがどの程度モデル内に組み込まれているかという指標でもある。

#### 【 0 0 8 2 】

サンプリング自体が、発生した非同期リアルタイム・イベントを最初の段階で検出する方法として認識できることに留意されたい。特定フライトが欠航になっているという航空会社からの指摘、又は販売促進キャンペーンの指摘等の非同期リアルタイム・イベントの明示的指摘に加えて又は代替的に、サンプリングを利用して、第3者が明示的に指摘しない非同期リアルタイム・イベントを暗示的に検出できる。この目的で、事前計算データベース・クエリ結果の全持ち分の実際の正確さを推定するために、全持ち分にわたって連続的又は規則的なサンプル処理を採用できる。このことにより、決定された実際の正確さと確率モデルが予測する正確さとの比較により、決定された実際の正確さが、確率モデルが予測する正確さよりも実質的に低いことが判明した場合、確率モデル内に反映されないイベントを暗示的に検出可能にする。

#### 【 0 0 8 3 】

検出されたイベントによって影響を受ける可能性がある1つ又は複数の持ち分の特定は、イベントの特性及びイベントについての知識に依存することを更に留意されたい。例えば、イベントが周期的サンプリング処理によって暗示的に検出されている場合、影響を受ける可能性がある持ち分は、同時に決定されている（即ち、実際の正確さを決定されたこれらの持ち分は、予測した正確さよりも実質的に低い）。一方、イベントの明示的シグナリングは、影響を受ける可能性がある事前計算データベース・クエリ結果の指摘を含むこともできる。例えば、航空会社が特定フライトの欠航をシグナリングした場合、欠航便前後の時間の間の同じ起点と目的地との間のフライトの価格は、影響を受ける可能性があると考えられる可能性がある。更に、過去の経験から、どの（明示的にシグナリングした）イベントがどの事前計算データベース・クエリ結果の正確さに影響を与える可能性があるかを知ることができる。例えば、どのリアルタイム・イベント及びリアルライフ・イベントが、連続サンプリング処理により検出された持ち分の正確さの低下に関与しているかを再追跡し、このようにして、いくつかの非同期リアルタイム・イベントの通常の影響についての知識を獲得することが可能である。

#### 【 0 0 8 4 】

次に、上記した、持ち分の実際の正確さ推定に対する分析、及び持ち分の実際の正確さと確率モデルが予測する正確さとの比較をより詳細に、より形式的に説明する。

#### 【 0 0 8 5 】

1つの事前計算データベース・クエリ結果  $x_i$  の実際の正確さを  $A c c_i$  と呼ぶ。事前計算データベース・クエリ結果  $x_i$  は、特定の時点で正確（ $A c c_i = 1$ ）、即ちその再計算結果に等しい、又は不正確（ $A c c_i = 0$ ）、即ちその再計算結果とは異なる、のいずれかである。全事前計算データベース・クエリ結果の収集を持ち分  $D_j$  に論理的に分割できる。以下の考慮は、1つの特定の例示的持ち分  $D$  に関連する。持ち分  $D$  に含まれ、確率モデルが予測する事前計算データベース・クエリ結果の平均正確さは、 $A c c_{m o d e l}(D)$  と示される。一方、 $D$  における実際に正確な事前計算データベース・クエリ結果の割合を  $A c c(D)$  と呼ぶ。

#### 【 0 0 8 6 】

上記のように、 $A c c(D)$  を決定又は推定するために、 $D$  における  $N$  個の事前計算データベース・クエリ結果のサンプル再計算を実施する。 $N$  個のうち正確な結果の数を  $A C_N$  と呼び、即ち  $N$  個のサンプル事前計算データベース・クエリ結果を再計算した後、 $A C_N$  個の再計算されたデータベース・クエリ結果は、それらの以前にキャッシュ化した対応

10

20

30

40

50

物と同一である一方で、 $(N - AC_N)$  個の再計算されたデータベース・クエリ結果は、それらの以前にキャッシュ化した対応物とは異なり、したがって不正確であることがわかっている。

【 0 0 8 7 】

上記のように、持ち分 D 内の 1 つの事前計算データベース・クエリ結果  $x_i$  に対する正確さの確率は、 $Acc(D)$  である。

【 0 0 8 8 】

このことは、二項式の法則  $AC_N \sim B(N, Acc(D))$  を採用することにより、D 内の N 個のサンプル事前計算データベース・クエリ結果に対して一般化できる。したがって、N 個のサンプルのうち K 個の事前計算データベース・クエリ結果が正確である確率は

10

【 0 0 8 9 】

【数 3】

$$P(AC_N = K) = \binom{N}{K} \times Acc(D)^K \times (1 - Acc(D))^{N-K}$$

である。

【 0 0 9 0 】

この法則の統計的期待値は、 $E(AC_N) = N \times Acc(D)$  である。

【 0 0 9 1 】

したがって、関係

20

【 0 0 9 2 】

【数 4】

$$\frac{AC_N}{N}$$

は、N 個の事前計算データベース・クエリ結果のサンプル事前計算の結果により伝えられる、持ち分 D の実際の正確さの推定、 $Acc(D)$  である。一方、 $Acc_{model}(D)$  は、確率モデルが示す  $Acc(D)$  の正確さの推定である。

【 0 0 9 3 】

所与のリアルタイム・イベントが確率モデル内に存在するか否かを決定するために、両方の値が互いに異なる、具体的には

30

【 0 0 9 4 】

【数 5】

$$\frac{AC_N}{N}$$

$< Acc_{model}(D)$  である場合の可能性があるケースでは、

【 0 0 9 5 】

【数 6】

$$\frac{AC_N}{N}$$

又は  $Acc_{model}(D)$  が  $Acc(D)$  のより信頼できる近似値であるかどうか判定することが必要である。この判定を準備するために、2 つの仮説  $H_0$  及び  $H_1$  を規定できる：

40

$H_0$  : 確率モデルの予測は、持ち分 D の正確さを適切に示し、即ち  $Acc(D) = Acc_{model}(D)$  である。

【 0 0 9 6 】

【数 7】

$$\frac{AC_N}{N}$$

は、サンプリング処理によって生じた統計的偏りのために  $Acc_{model}(D)$  よりも小さい（即ち、偶発的に、選択された N 個の事前計算データベース・クエリ結果は、D に

50

における全事前計算データベース・クエリ結果の平均、及びしたがってN個のサンプルが  $A_{cc}(D)$  を見本として反映しないという点でより正確である)。

$H_1$  : 確率モデルにより予期されなかった何かが発生し、したがって  $A_{cc\_model}(D)$  は、 $A_{cc}(D)$  の不適切な推定である。むしろ、サンプル再計算によって得られた実際の正確さ推定は、最適な利用可能な推定であり、即ち  $A_{cc}(D)$

【0097】

【数8】

$$\frac{AC_N}{N}$$

である。

10

【0098】

$H_0$  が真であるという判定  $D_0$ 、又は  $H_1$  が真であるという判定  $D_1$  の実施は、以下の方法で行うことができる：

【0099】

実際の正確さが予測した正確さよりも所与の程度を超えて低い場合、確率モデルが予測する正確さが不適切である可能性は増大し、低減させる必要がある。例えば、 $A_{cc\_model}(D)$  は85%の正確さ  $A_{cc}(D)$  を示す一方で、サンプル再計算が受信した実際の正確さは、Dにおける正確な事前計算データベース・クエリ結果の20%のみを示す場合がそうである。この場合、 $D_0$  が正しい判断である。しかし、サンプリングによって受信された実際の正確さ及びモデルが予測した正確さが類似する場合、即ち、両方の指摘の間の偏りが比較的小さい場合、確率モデルが予測する正確さがおそらく適切である。一例を挙げれば、 $A_{cc\_model}(D)$  が85%の正確さ  $A_{cc}(D)$  を示し、

20

【0100】

【数9】

$$\frac{AC_N}{N}$$

がNが80%の正確さに達する場合がそうであり得る。この場合、 $D_1$  が正しい判定である。

【0101】

$D_0$  と  $D_1$  との間の境界を規定するために、閾値  $K_{Th}$  を導入する：

30

【0102】

$AC_N > K_{Th}$  である場合、確率モデルが予測する正確さ  $A_{cc\_model}(D)$  は、 $A_{cc}(D)$  の適切な近似値として考慮され、 $D_0$  が選択される。 $AC_N \leq K_{Th}$  である場合、確率モデルが予測する正確さ  $A_{cc\_model}(D)$  は、 $A_{cc}(D)$  の適切な近似値として考慮されず、 $D_1$  が選択される。

【0103】

$K_{Th}$  を適切な値に設定するために、確率モデルの質についての情報を有することが有利である。そのような情報は、通常、モデルを展開する前及び/又はモデルの継続使用中に実施されていたモデル検証処理から入手可能である。例えば、確率モデルの1年の使用の間に、確率モデルによって反映されない非同期リアルタイム・イベントがかなりまれであり、例えば365日のうち3日だけ発生することが観察されていた場合がある。この場合、 $A_{cc\_model}(D)$  が時間の99%について適切であり(=  $H_0$ )、時間の1%のみについて不適切である(=  $H_1$ )ことが仮定できる。別の例では、確率モデル内に含まれない非同期リアルタイム・イベントが1年のうち180日等、比較的頻繁に発生する場合がある。したがって、 $A_{cc\_model}(D)$  は1日おきのケースのみで適切である、即ち時間の最大50%で適切であり(=  $H_0$ )、それ以外の場合は不適切である(=  $H_1$ )ことが仮定できる。

40

【0104】

確率モデルの過去の質/確実さを考慮することによって  $K_{Th}$  を設定する1つの例示的方法は、誤りの判定  $D_1$  (即ち、 $A_{cc\_model}(D)$  は、実際には適切な推定である

50

にもかかわらず、 $Acc(D)$ の適切な推定であるとみなされない)の確率を考慮することである。確率モデルの予測が正しい確率は：

【 0 1 0 5 】

【数 1 0 】

$$P(AC_N = K | H_0) = \binom{K}{N} \times Acc_{model}(D)^K \times (1 - Acc_{model}(D))^{N-K}$$

として表すことができる。

【 0 1 0 6 】

仮説  $H_0$  が真であり、誤りの判定  $D_1$  が取られる確率は、 $AC_N$  が  $K_{Th}$  よりも低い確率：

【 0 1 0 7 】

【数 1 1 】

$$\alpha = P(D_1 | H_0) = P(AC_N \leq K_{Th} | H_0) = \sum_{K \leq K_{Th}} \binom{K}{N} \times Acc_{model}(D)^K \times (1 - Acc_{model}(D))^{N-K}$$

である。

【 0 1 0 8 】

この式は、確率モデルの過去の確実さに応じて  $\alpha$  を選択し、 $K_{Th}$  を規定可能にする。例えば、確率モデルが時間の 99% に関して正しかった場合、 $\alpha$  は 0.01 に設定される。次に、 $K_{Th}$  の閾値を以前の式を使用して計算できる。K 個の正確なサンプル再計算結果の数が  $K_{Th}$  よりも低い場合、即ち、サンプル再計算によって決定した実際の正確さの推定

【 0 1 0 9 】

【数 1 2 】

$$\frac{AC_N}{N}$$

は、 $Acc_{model}(D)$  よりも確実な  $Acc(D)$  の推定としてみなされる：

持ち分Dの実際の正確さを推定するためのサンプル再計算の数N	N=10	N=20	N=50	N=100
$Acc_{model}=0.85$ である場合に確率モデルが正確であると予測するN個のサンプル事前計算データベース・クエリ結果の数K	K=8	K=17	K=42	K=85
閾値 $K_{Th}$ ( $\alpha=0.01$ の場合)	6	14	36	76
イベントが確率モデル内の見本となると考慮されない $AC_N$ (正確なサンプル再計算) のための範囲 ( $\alpha=0.01$ 及び対応する $K_{Th}$ の場合)	$AC_N \leq 5$	$AC_N \leq 13$	$AC_N \leq 35$	$AC_N \leq 75$

【 0 1 1 0 】

したがって、要約すると、 $AC_N$  が  $K_{Th}$  以上である場合、非同期リアルタイム・イベントは、確率モデル内の見本となると考慮され、それぞれの事前計算データベース・クエリ結果の確率の修正は必要ではない。しかし、 $AC_N$  が  $K_{Th}$  を下回る場合、イベントは確率モデル内の見本とならないことが仮定される。

【 0 1 1 1 】

例えば販売促進キャンペーン又は整備イベントのような、予測可能ではないイベント、したがって確率モデル内に全く含まれないイベントに関して、イベントは、できるだけ早く処理される。一般に、このことは、特定のリアルタイム・イベントによって影響を受ける可能性があるこうした事前計算データベース・クエリ結果の確率が、修正、具体的には減少されることを意味する。イベントが確率モデル内に存在するか否かを決定する上記例を続けると、この修正は、サンプル再計算によって獲得した  $Acc(D)$  の推定、即ち

【 0 1 1 2 】

【数 1 3】

$$\frac{AC_N}{N}$$

を持ち分 D における古くなった事前計算データベース・クエリ結果の確率に適用することによって任意選択で達成される。例えば、

【0 1 1 3】

【数 1 4】

$$\frac{AC_N}{N}$$

が N が  $Acc_{model}(D)$  よりも 25% 低いことがわかっている場合、持ち分 D における事前計算データベース・クエリ結果のための確率モデルの現在の正確さ断定は、例えば同様に 25% 減少する。この例示的な修正は、図 4 b によって可視化される（図 4 の x 軸及び y 軸は、図 4 a と同じ意味を有する。即ち、x 軸は最終再計算（= 寿命 t）以来の時間単位での時間を示し、y 軸は正確さ / 古さの確率を示す）。

10

【0 1 1 4】

この修正を行った後、どの事前計算データベース・クエリ結果を再計算すべきかという判定が通常通り、即ち、同様に確率の修正が起こっていないかのように取られる。上記で概説した例示的履歴統計モデルを再度参照すると、正確さ / 古さに関する確率モデルが示す確率は、 $e^{-\lambda_i t_i}$  である。しかし、この確率は、例えば上記で詳細に説明した方策を適用することにより、認識した非同期リアルタイム・イベントに応答して修正できるので、この確率は、実際には、時間  $t_n$  における推定正確さ / 古さ  $Acc_i^{t_n}$  が時間  $t_{n-1}$  におけるその以前の値から計算されるステップ：

20

【0 1 1 5】

【数 1 5】

$$Acc_i^{t_n} = Acc_i^{t_{n-1}} \times e^{-\lambda_i(t_n - t_{n-1})}$$

の後に計算されるステップであってもよい。

【0 1 1 6】

分析器 6 による次の正確さ計算は、やはり以前の値（リアルタイム・イベントに応答して修正された可能性がある）から開始する。

30

【0 1 1 7】

到来する非同期リアルタイム・イベントに応答する確率の修正、例えば確率モデルが予測する正確さは、それぞれの事前計算データベース・クエリ結果が MCP 3 によって次に再計算された後に再設定される。

【0 1 1 8】

運賃又は利用可能性の変更のような、少なくとも特定の程度に確率モデル内の見本となるイベントは、任意選択で蓄積され、イベントの出現は、確率モデルの予測と定期的に比較される。イベント・ピークがモデルと局所的に一致しないとき、即ち、イベントの突発的な逸脱が確率モデルの基礎をなす統計から著しく外れる場合、影響を受けた価格を古くなった可能性があるとしてマークし、できるだけ早くそれらの価格を再計算するようにする。このことにより、確率モデル内に既に存在する、したがって分析器 6 が予め行った決定により既に考慮されたイベントにより生じた「ノイズ」を除外する。

40

【0 1 1 9】

任意選択で、到来する非同期リアルタイム・イベントの扱いは、統合器 7 から機能的に独立した、又は統合器 7 の副構成要素である専用の構成要素、イベント・マネージャ・モジュールによって行う。イベントは、入力マネージャ 5 によって受信され、内部データ・リポジトリに集められることなく最初に記憶される。イベント・マネージャは、到来する非同期リアルタイム・イベントがあれば受信し、例えば上記で詳細に説明した方策を適用することによりイベントを処理し、可能性としては影響を受けた事前計算データベース・クエリ結果が古くなった / 不正確である確率の修正を実施する。任意選択で、上記した方

50

策を所与の数のイベントがイベント・マネージャによって認識されている場合にのみ適用し、次に、それに従って必要な場合にのみ各事前計算データベース・クエリ結果の推定正確さを修正する。

【 0 1 2 0 】

任意選択で、最適化の目的のために、統合器 7 は、内部データ見本 8 の格子面 ( grid - view ) に作用し、即ち、統合器 7 は、孤立した様式で価格のセットに作用するのではなく、そのアルゴリズムの間で互いに隣接する価格の群を考慮する。このアプローチでは、隣接する価格データ・セットは、集合性質の値を有する単一のデータ・セットとしてみられる。集合データ・セットに作用すると、疎な再計算指令の生成を制限し、したがって M C P 3 における相互化及び最適化の機会が増大する。このことは、計算費用を低減する。

10

【 0 1 2 1 】

図 5 は、上記の詳細な説明を要約し、本明細書で提示するキャッシュ更新方法の概要を示す。

【 0 1 2 2 】

事前計算データベース・クエリ結果のキャッシュを最新に保持する処理は、キャッシュ化クエリ結果の不正確さの確率の決定 ( 1 4 ) から開始する。この決定は、2つの論理レベルに位置する2つの行動から構成される。第1に及び一般に、統計及び確率データに基づく予測モデルは、特定のキャッシュ化クエリ結果が ( 仮説的に ) 再計算したクエリ結果に対応しない可能性を推定するために採用される。第2に、及びより具体的には、キャッシュ化クエリ結果が古くなる確率に影響を与える可能性があり、その確率が増大する可能性があるリアルタイム・イベントをそれぞれ考慮に入れる。こうしたリアルタイム・イベントは、リアルタイム・イベントが一般に特定のキャッシュ化クエリ結果の不正確さを確実に示すのではなく、この点において非確定的であることを特徴とする。したがって、リアルタイム・イベント発生時に、正確さ及び不正確さそれぞれの可能性に対する確率的な結論のみを出すことができる。

20

【 0 1 2 3 】

キャッシュ化データベース・クエリ結果が古くなったことが決定される確率に基づいて、D C P 2 がデータベース再計算指令を自動的に発行する ( 1 5 )。こうした命令は、M C P 3 によって受信され、M C P 3 は、次に、それぞれの結果を再計算し、結果を D C P 2 に返信する ( 1 6 )。今度は、D C P 2 は、結果を受信し、結果をローカル見本部 8 に記憶する ( 1 7 )。これにより、1つの更新サイクルを終わりにし、次のサイクルが確率の決定 ( 1 4 ) で再反復される。

30

【 0 1 2 4 】

次に、本発明のキャッシュ更新方策の手順のタイミングに関する特定の例を図 6 に関して説明する。この例では、D C P 2 は、再計算指令を 2 0 分ごとに生成するように構成され、即ち、キャッシュ化データが古くなったかどうかの確率を決定し、再計算指令を生成、発行するラウンド又はサイクルは、2 0 分かかる。M C P 3 におけるリソースは、一日中アプリオリに知られ、D C P 2 は、M C P 3 で利用可能な計算リソースを認識し、したがって再計算の量を M C P 3 の利用可能なリソースと同期できる。

40

【 0 1 2 5 】

再計算サイクルの初めに、D C P 2 は、内部データベース 8 に記憶したキャッシュ化データベース・クエリ結果、即ち価格付き旅行推奨の現在の正確さを分析する。ラウンドは、2 0 分のラウンド終了時に M C P 3 により処理される一組の再計算指令を生じさせることになる。一方、M C P 3 側では、最終サイクルからの指令が計算され、新たな価格推奨が生成され、D C P に返信され、この D C P で、新たな価格推奨は、記憶され、次のサイクルで繰り返される情報の分析及び更新のために利用可能である。

【 0 1 2 6 】

図 6 は、M C P が 0 4 : 0 0 から 0 5 : 0 0 a . m . の間の時間区間で著しい利用可能リソースを有し、そのため、多くの再計算をこの時間に実施できることを示す。しかし、

50



その後、利用可能リソースは9:00 a.m.までなく、そのため、再計算はその時間では可能ではない。その日の後、09:00 a.m.から7:00 p.m.までにいくつかのリソースがMCP3で利用可能である。

【0127】

04:20 a.m.でサイクルが開始される間、分析器6は、キャッシュの正確さを分析する一方で、統合器7は、それに応じて再計算指令を生成する。こうした指令は、MCP3によって04:40 a.m.に実施されることになる。分析器6は、MCPがラウンドの開始時に受信したMCPの価格推奨に注目する。分析器6は、受信した価格と、価格の値が内部リポジトリに記憶されている以前の価格との間の差異を計数する。差異に基づき、分析器6は、繰り返される「揮発性」情報ソースを修正する。入力マネージャ5は、受信したMCP価格を更なる検査のために保存する。

10

【0128】

4:40から5:00 a.m.のサイクルでは、MCP3は、DCP2から受信した再計算指令を区間04:20から4:40 a.m.の間に処理する。DCP2は、DCP2が、到来する時間スライス(05:00 a.m.)の間及び09:00 a.m.までのその次の時間スライスの間、いかなる再計算指令も生成できないことを認識している。しかし、DCP2は、データ・モデルを依然として連続的に分析して全キャッシュの価格付き旅行推奨の現在の正確さを更新する。DCP2は、08:40 a.m.までの将来のサイクルごとと同じことを行うことになる。

【0129】

20

08:40 a.m.では、分析器6は、再計算を全く行わない前の4時間の間にキャッシュの正確さが低下したことを決定する。分析器6は、それに応じて以下のサイクルにわたって再計算指令を生成するが、09:00 a.m.から7:00 p.m.までのMCP3における利用可能リソースの量の限度のためにより少ない程度にだけ生成する。次に、09:00 a.m.では、MCP3は、MCP3が前の区間(即ち08:40から09:00 a.m.)で受信した新たな再計算指令を処理し始め、6:40から7:00 p.m.まで続くラウンドの終了後、停止することになる。

【0130】

その後、MCP3では夜間を通して更なる利用可能リソースはない。したがって、DCP2は、更なる再計算指令を何も生成しないが、確率モデル及び可能性としては到来するリアルタイム・イベントに基づきキャッシュの正確さを分析し続けることになる。

30

【0131】

最後に、図7は、図2のキャッシュ・プラットフォーム2の機能を提供するコンピュータ・システムの概略図である。キャッシュ・プラットフォーム2内では、コンピュータ・システムに本明細書で論じる方法のいずれかを実施させる一組の命令を実行できる。キャッシュ・プラットフォーム2は、バス104を介して互いに通信するプロセッサ101、メイン・メモリ102及びネットワーク・インターフェース・デバイス103を含む。任意選択で、キャッシュ・プラットフォーム2は、スタティック・メモリ105及びディスク・ドライブ・ユニット106を更に含むことができる。ビデオ画面107、英数字入力デバイス108及びカーソル制御デバイス109は、配信リスト・ナビゲータ・ユーザ・インターフェースを形成できる。ネットワーク・インターフェース・デバイス103は、データ・キャッシュ・プラットフォーム2を計算プラットフォーム3、統計サーバ9等の予測モデルを満たすのに必要な統計データのリソース、揮発性データベース10及び最初の正確さデータベース11、リアルタイム・イベントのリソース、インターネット及び/又はあらゆる他のネットワークに接続する。上記した方法のいずれか1つ又は全てを具体化する命令のセット(即ちソフトウェア)110は、完全に又は少なくとも部分的に機械可読媒体、例えばメイン・メモリ102及び/又はプロセッサ101内に又はそれに接して存在する。ソフトウェア110が存在する機械可読媒体は、ディスク・ドライブ・ユニット106の一部である不揮発性データ・キャリア111(例えば取外し不可能磁気ハード・ディスク又は取外し可能光ディスク又は磁気ディスク)であってもよい。ソフトウェ

40

50

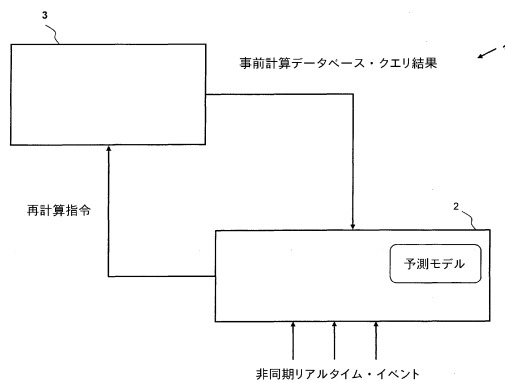
ア 1 1 0 は、伝搬信号 1 1 2 としてインターネットを介し、ネットワーク・インターフェースデバイス 1 0 3 を通して更に送信又は受信できる。

# 【 0 1 3 2 】

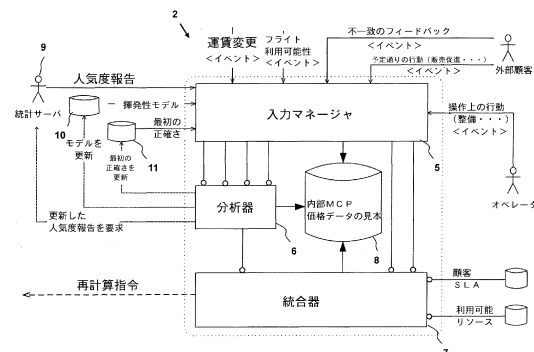
本発明のキャッシュ更新方策は、キャッシュ再計算判定を自動的に生成する手段を提供する。キャッシュ再計算判定を自動的に生成する手段は、どのキャッシュ化クエリ結果を再計算すべきか決定し、計算プラットフォームで利用可能計算リソースを考慮することにより再計算を時間的にも制御する。したがって、一般に、キャッシュ化クエリ結果の正確さは、最新さ及び古さのそれぞれを経時的にモデル化する確率モデルに基づき推定される。この古さの分析は、1時間あたり、キャッシュ化データベース・クエリ結果の再計算の基礎をなす数十億のデータ・セットの処理を可能にする。

10

【 図 1 】

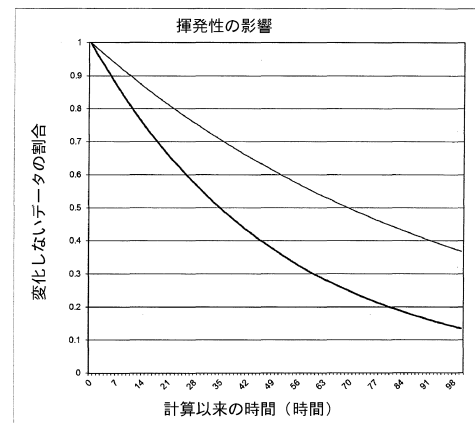
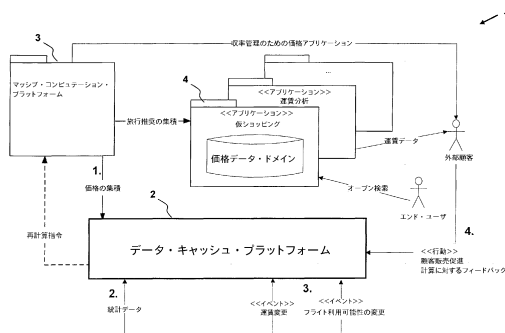


【 図 3 】

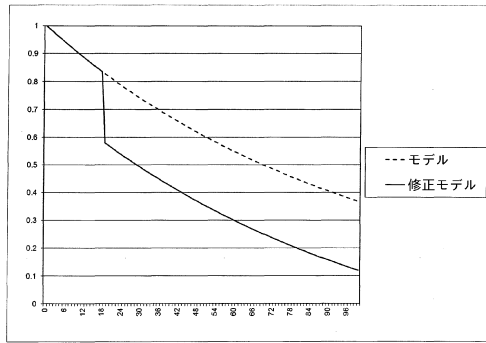


【 図 4 a 】

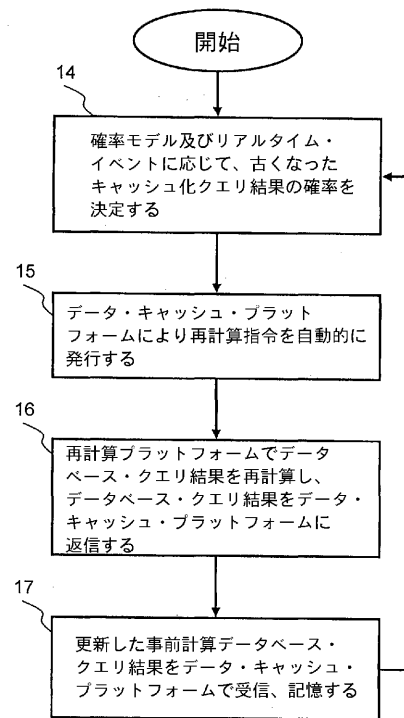
【 図 2 】



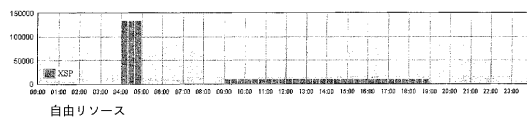
【図 4 b】



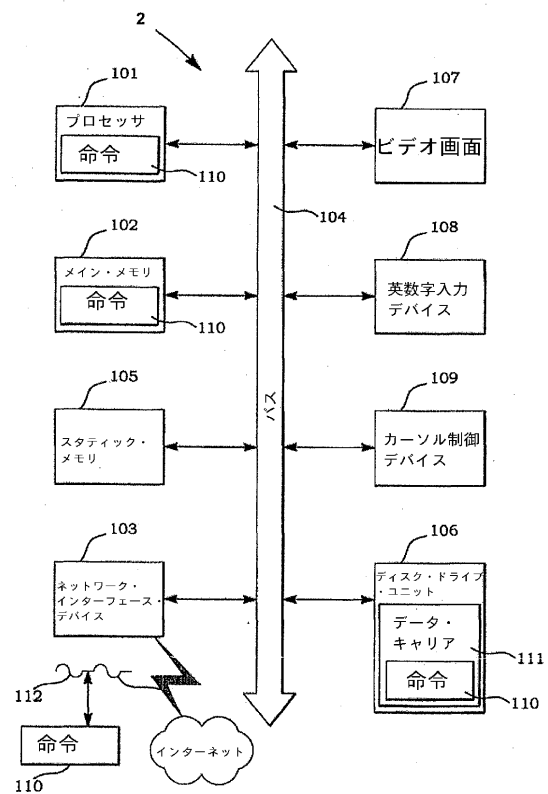
【図 5】



【図 6】



【図 7】



## フロントページの続き

(74)代理人 100133400

弁理士 阿部 達彦

(72)発明者 シャブリニ, ダミエン

フランス国 エフ - 0 6 5 2 0 マガグノスク シュマン デ シャシュー アルパン 3 1

(72)発明者 レグランド, ギオーム

フランス国 エフ - 0 6 1 3 0 グラッセ シュマン デ コムテス 1 1 ヴィラ 4 7 ドメン デ オリヴィエ

(72)発明者 ジヤナン, ベノワ

フランス国 エフ - 0 6 4 1 0 ピオ ドメン デ クロゾンヌ アヴェニュー デ パン 4 9 7

(72)発明者 イスナルディ, ルーク

フランス国 エフ - 0 6 8 0 0 カンヌ スル メル ルー ミシェル アンジ 1 8

(72)発明者 マイヨ, ニコラス

フランス国 エフ - 6 4 1 0 ピオ アヴェニュー デ フォヴェット 7 8 6

(72)発明者 ロベリン, シャール - アントワン

フランス国 エフ - 0 6 6 0 0 アンチーブ アヴェニュー シエー 1

(72)発明者 ダニエロ, ルディ

フランス国 エフ - 0 6 2 0 0 ニース シュマン デ ラ トゥール デ ベレ ビス 1 8

審査官 樋口 龍弥

(56)参考文献 国際公開第 2 0 0 6 / 0 4 6 4 8 6 (WO, A 1)

米国特許出願公開第 2 0 0 8 / 0 1 4 1 2 6 1 (US, A 1)

特開 2 0 1 1 - 1 8 6 9 0 3 (JP, A)

特表 2 0 0 4 - 5 0 4 6 8 1 (JP, A)

(58)調査した分野(Int.Cl., DB名)

G 0 6 F 1 7 / 3 0

G 0 6 F 1 2 / 0 0