

(19) 日本国特許庁(JP)

(12) 特 許 公 報(B2)

(11) 特許番号

特許第5282908号  
(P5282908)

(45) 発行日 平成25年9月4日(2013.9.4)

(24) 登録日 平成25年6月7日(2013.6.7)

(51) Int.Cl. F I  
**G06F 11/34 (2006.01)** G O 6 F 11/34 S  
**G06F 11/36 (2006.01)** G O 6 F 9/06 6 2 O R

請求項の数 8 (全 22 頁)

(21) 出願番号	特願2009-535996 (P2009-535996)	(73) 特許権者	000004237
(86) (22) 出願日	平成20年8月21日 (2008. 8. 21)		日本電気株式会社
(86) 国際出願番号	PCT/JP2008/064871		東京都港区芝五丁目7番1号
(87) 国際公開番号	W02009/044589	(74) 代理人	100102864
(87) 国際公開日	平成21年4月9日 (2009. 4. 9)		弁理士 工藤 実
審査請求日	平成23年7月14日 (2011. 7. 14)	(72) 発明者	佐々木 盛朗
(31) 優先権主張番号	特願2007-259947 (P2007-259947)		東京都港区芝五丁目7番1号 日本電気株式会社内
(32) 優先日	平成19年10月3日 (2007. 10. 3)		
(33) 優先権主張国	日本国 (JP)	審査官	坂庭 剛史

最終頁に続く

(54) 【発明の名称】 階層型負荷推定システム、方法およびプログラム

(57) 【特許請求の範囲】

【請求項1】

多層システムにおいて階層的に呼び出される階層処理の設計を示す階層呼出情報と、前記階層処理のうちデータベース操作処理によって発行されるデータベース操作文を規定する発行情報と、データベース操作文あたりの負荷を示す負荷情報とが格納される記憶装置と、

前記階層呼出情報を参照して、前記データベース操作処理の呼出回数を算出する呼出回数算出モジュールと、

前記算出された呼出回数と前記発行情報を参照して、前記データベース操作処理によって発行されるデータベース操作文の発行回数を算出する発行回数算出モジュールと、

前記算出された発行回数と前記負荷情報を参照して、前記データベース操作処理によって発行されるデータベース操作文の負荷を算出する負荷算出モジュールと

を備える

階層型負荷推定システム。

【請求項2】

請求項1に記載の階層型負荷推定システムであって、

前記発行情報は、データベース操作文の発行回数を前記データベース操作処理毎に示し、

前記発行回数算出モジュールは、前記データベース操作処理によって発行されるデータベース操作文の発行回数の合計を算出し、

前記負荷算出モジュールは、前記データベース操作処理によって発行されるデータベース操作文の負荷の合計を算出する  
階層型負荷推定システム。

【請求項 3】

請求項 1 に記載の階層型負荷推定システムであって、  
前記発行情報は、データベース操作文の分類を前記データベース操作処理毎に示し、  
前記発行回数算出モジュールは、前記発行回数を前記分類毎に算出し、  
前記負荷情報は、データベース操作文あたりの負荷を前記分類毎に示し、  
前記負荷算出モジュールは、前記負荷を前記分類毎に算出する  
階層型負荷推定システム。

10

【請求項 4】

請求項 1 に記載の階層型負荷推定システムであって、  
前記発行情報は、データベース操作文の分類を前記データベース操作処理及びエンティティ毎に示し、  
前記発行回数算出モジュールは、前記発行回数を前記分類及び前記エンティティ毎に算出し、  
前記負荷情報は、データベース操作文あたりの負荷を前記分類及び前記エンティティ毎に示し、  
前記負荷算出モジュールは、前記負荷を前記分類及び前記エンティティ毎に算出する  
階層型負荷推定システム。

20

【請求項 5】

請求項 2 乃至 4 のいずれか一項に記載の階層型負荷推定システムであって、  
前記多層システムは、データベース層を含み、前記負荷算出モジュールは、前記負荷を前記データベース層にかかる負荷として算出する  
階層型負荷推定システム。

【請求項 6】

請求項 1 乃至 5 のいずれか一項に記載の階層型負荷推定システムであって、  
前記階層呼出情報は、  
前記階層処理のうち外部から要求される最上位処理の呼出回数を示す第 1 情報と、  
前記階層処理における上位処理と下位処理との間の呼び出し関係を規定する第 2 情報と  
を含み、  
前記呼出回数算出モジュールは、前記第 1 情報と前記第 2 情報を用い、各層の処理の呼出回数を上位層から下位層に向けて順番に算出することによって、前記データベース操作処理の呼出回数を算出する  
階層型負荷推定システム。

30

【請求項 7】

コンピュータが実行する階層型負荷推定方法であって、  
多層システムにおいて階層的に呼び出される階層処理の設計を示す階層呼出情報を記憶装置から読み出すことと、  
前記階層処理のうちデータベース操作処理によって発行されるデータベース操作文を規定する発行情報を前記記憶装置から読み出すことと、  
データベース操作文あたりの負荷を示す負荷情報を前記記憶装置から読み出すことと、  
前記階層呼出情報を参照して、前記データベース操作処理の呼出回数を算出することと  
、  
前記算出された呼出回数と前記発行情報を参照して、前記データベース操作処理によって発行されるデータベース操作文の発行回数を算出することと、  
前記算出された発行回数と前記負荷情報を参照して、前記データベース操作処理によって発行されるデータベース操作文の負荷を算出することと  
を含む  
階層型負荷推定方法。

40

50

## 【請求項 8】

コンピュータ読み取り可能な記録媒体に記録された階層型負荷推定プログラムであって

、  
多層システムにおいて階層的に呼び出される階層処理の設計を示す階層呼出情報を記憶装置から読み出すことと、

前記階層処理のうちデータベース操作処理によって発行されるデータベース操作文を規定する発行情報を前記記憶装置から読み出すことと、

データベース操作文あたりの負荷を示す負荷情報を前記記憶装置から読み出すことと、  
前記階層呼出情報を参照して、前記データベース操作処理の呼出回数を算出することと

10

、  
前記算出された呼出回数と前記発行情報を参照して、前記データベース操作処理によって発行されるデータベース操作文の発行回数を算出することと、

前記算出された発行回数と前記負荷情報を参照して、前記データベース操作処理によって発行されるデータベース操作文の負荷を算出することと

をコンピュータに実行させる

階層型負荷推定プログラム。

## 【発明の詳細な説明】

## 【技術分野】

## 【0001】

本発明は、システムにおける負荷を推定するための技術に関する。特に、本発明は、多層システムにおける階層型負荷を推定するための技術に関する。

20

## 【背景技術】

## 【0002】

システムの構築は、設計フェーズ、開発フェーズ、運用フェーズの三つのフェーズに分けられる。一般に、設計フェーズでは機能が、開発フェーズでは実装が、運用フェーズでは性能が主要な問題となる。これらのフェーズは、新規システムを構築するときだけでなく、既存システムを変更するときにも現れる。

## 【0003】

運用フェーズにおいてシステムが十分な性能を発揮するためには、支障なく機能を提供できるだけの十分な計算機資源が必要である。必要な計算機資源の量は、機能を提供する際に発生する負荷の量に依存する。少ない負荷を発生させる機能は少ない計算機資源で提供でき、多くの負荷を発生させる機能は多くの計算機資源を必要とする。従って、システム性能を評価するためには、システム負荷を測定することが望ましい。

30

## 【0004】

運用フェーズにおいてシステム負荷を測定するための技術として、次のものが知られている。例えば、文献“gprof: A Call Graph Execution Profiler, by S. Graham, P. Kessler and M. Mckusick, Proceedings of the 1982 SIGPLAN Symposium on Compiler Construction, Vol. 17, No. 6, pp. 120-126, 1982.”に開示されているプロファイリング技術を用いれば、プログラム中の個々の関数の呼出回数をカウントすることができる。また、特開2005-190277号公報には、サーバープログラム間の通信を観測することで得られるIPアドレスとポート番号から、サーバープログラムの呼出頻度をカウントする方法が記載されている。

40

## 【0005】

運用フェーズにおいてシステム性能の不足が発覚したとき、十分な性能を達成するために設計等を見直さなければならない場合がある。しかしながらこの場合、多大な手戻りの工数が発生し、システム構築コストが増大する。逆に言えば、もし上流フェーズの段階でシステム性能やシステム負荷を推定することができれば、手戻り工数を抑制し、コスト増大を抑制することができる。特に、設計フェーズの段階でシステム負荷を推定することは

50

、システム構築コストの削減に有効である。

【0006】

しかしながら、設計フェーズにおいて従来使用できた技術は、主にシステムの機能設計に関する技術であった。例えば、文献“ALLFusion(登録商標) ERwin(登録商標) Data Modeler r7.”には、データベースの機能設計ツールとしての「ERwin(登録商標)」が紹介されている。このツールは、論理設計の物理設計への変換、個別処理の最適化、既存のデータベースのリバースエンジニアリング等の機能を提供している。また、特開平8 202541号公報には、DFD(Data Flow Diagram)の変更とER(Entity Relationship)の変更を相互に反映させる方法が記載されている。しかし、これらのツールや方法では、設計フェーズにおいてシステム負荷を推定

10

【0007】

近年、階層構造を有する「多層システム」が増加している。多層システムでは、処理内容の頻繁な変更に対応できるように、また、性能を容易に向上させられるように、サービス要求は複数のサーバー層でパイプライン的に処理される。典型的な多層システムとしては、ウェブ層で表示処理を、アプリケーション層で演算処理を、データベース層でデータ処理を行う三層システムが知られている。このような多層システムの性能は、複数のサーバー層のうちボトルネック層の性能によって主に決定される。例えば上記三層システムの場合、一般的にはデータベース層がボトルネックとなり、そのデータベース層の性能が三層システムの性能を決定する。従って、三層システムの性能を定量的に推定するためには

20

、ボトルネック層となりがちなデータベース層にかかる負荷を定量的に推定することが好ましい。

【0008】

データベース層には、階層的な負荷(以下、「階層型負荷」と参照される)がかかる。多層システムでは、上位層の処理が下位層の処理を呼び出す。つまり、サービス要求によって呼び出される処理が他の処理を呼び出し、呼び出された処理が更に他の処理を呼び出し、というように処理が階層的に呼び出される。そのような処理は、以下「階層処理」と参照される。階層型負荷とは、このように階層的に呼び出される階層処理によって生じる負荷である。データベース層は、通常は多層システムの最下位層である。従って、データベース層には階層型負荷がかかる。

30

【0009】

また、データベース層への処理要求のほとんどは、「DML(Data Manipulation Language)文」で発行される。DML文は、データベースに対する問い合わせに使用されるSQL(Structured Query Language)文の一部である。SQL文は大きくわけて、DML文、DDL(Data Definition Language)文、DCL(Data Control Language)文の三種類に分類されるが、ほとんどの負荷はDML文の処理によって生じる。DML文としては、データ抽出を指示する“select文”、データ更新を指示する“update文”、データ挿入を指示する“insert文”、データ削除を指示する“delete文”などが挙げられる。これらDML文には、データが操作されるテーブル、つまり、アクセス対象のデータセットが明記される。尚、アクセスするデータ量が同じであっても、DML文の種類によって処理内容と処理量は異なる。例えば、select文ではデータ読み出し処理だけが発生するが、update文ではデータ書き込みだけでなく、データロックやログの書き込み等の処理が発生する。

40

【発明の開示】

【0010】

本発明の1つの目的は、多層システムにおける階層型負荷を設計フェーズにおいて定量的に推定することができる技術を提供することにある。

【0011】

本発明の第1の観点において、階層型負荷推定システムが提供される。その階層型負荷推定システムは、記憶装置と、呼出回数算出モジュールと、発行回数算出モジュールとを

50

備える。記憶装置には、多層システムにおいて階層的に呼び出される階層処理の設計を示す階層呼出情報と、階層処理のうちデータベース操作処理によって発行されるデータベース操作文を規定する発行情報とが格納される。呼出回数算出モジュールは、階層呼出情報を参照して、データベース操作処理の呼出回数を算出する。発行回数算出モジュールは、算出された呼出回数と発行情報を参照して、データベース操作処理によって発行されるデータベース操作文の発行回数を算出する。

【0012】

本発明の第2の観点において、階層型負荷推定方法が提供される。その階層型負荷推定方法は、多層システムにおいて階層的に呼び出される階層処理の設計を示す階層呼出情報を記憶装置から読み出すことと、階層処理のうちデータベース操作処理によって発行されるデータベース操作文を規定する発行情報を記憶装置から読み出すことと、階層呼出情報を参照して、データベース操作処理の呼出回数を算出することと、算出された呼出回数と発行情報を参照して、データベース操作処理によって発行されるデータベース操作文の発行回数を算出することと、を含む。

10

【0013】

本発明の第3の観点において、階層型負荷推定プログラムが提供される。その階層型負荷推定プログラムは、多層システムにおいて階層的に呼び出される階層処理の設計を示す階層呼出情報を記憶装置から読み出すことと、階層処理のうちデータベース操作処理によって発行されるデータベース操作文を規定する発行情報を記憶装置から読み出すことと、階層呼出情報を参照して、データベース操作処理の呼出回数を算出することと、算出された呼出回数と発行情報を参照して、データベース操作処理によって発行されるデータベース操作文の発行回数を算出することと、をコンピュータに実行させる。

20

【0014】

本発明によって算出されたデータベース操作文の発行回数から、データベース層にかかる階層型負荷を見積もることができる。つまり、多層システムにおける階層型負荷を定量的に推定することができる。その階層型負荷の推定において、実際にシステムを構築することは不要である。たとえ設計フェーズの段階であっても、階層型負荷を定量的に推定することが可能となる。これにより、運用フェーズから設計フェーズへの戻りが防止され、システム構築コストが削減される。

【図面の簡単な説明】

30

【0015】

上記及び他の目的、長所、特徴は、次の図面と共に説明される本発明の実施の形態により明らかになるであろう。

【0016】

【図1】図1は、多層システムにおける階層処理及び本発明の実施の形態において用いられる情報を説明するための概念図である。

【図2】図2は、本発明の実施の形態に係る階層型負荷推定システムの構成例を示すブロック図である。

【図3】図3は、外部呼出回数情報の一例を示す概念図である。

【図4A】図4Aは、プロシージャ呼出情報の一例を示す概念図である。

40

【図4B】図4Bは、プロシージャ呼出情報の他の例を示す概念図である。

【図5】図5は、DML発行情報の一例を示す概念図である。

【図6】図6は、DML負荷情報の一例を示す概念図である。

【図7】図7は、本発明の実施の形態に係る階層型負荷システムによる処理フローを示すブロック図である。

【図8】図8は、本発明の実施の形態に係る階層型負荷システムによる処理フローを示すフローチャートである。

【図9】図9は、ステップS100を示すフローチャートである。

【図10】図10は、ステップS140の一例を示すフローチャートである。

【図11】図11は、ステップS200を示すフローチャートである。

50

【図12】図12は、第1及び第2の処理例における外部呼出回数情報を示す概念図である。

【図13】図13は、第1及び第2の処理例におけるプロシージャ呼出情報を示す概念図である。

【図14】図14は、第1の処理例におけるDML発行情報を示す概念図である。

【図15】図15は、第1の処理例におけるDML負荷情報を示す概念図である。

【図16】図16は、第1の処理例における最下位呼出回数情報を示す概念図である。

【図17】図17は、第1の処理例における発行回数情報を示す概念図である。

【図18】図18は、第1の処理例における推定負荷情報を示す概念図である。

【図19】図19は、第2の処理例におけるDML発行情報を示す概念図である。

10

【図20】図20は、第2の処理例におけるDML負荷情報を示す概念図である。

【図21】図21は、第2の処理例における発行回数情報を示す概念図である。

【図22】図22は、第2の処理例における推定負荷情報を示す概念図である。

【発明を実施するための最良の形態】

【0017】

添付図面を参照して、本発明の実施の形態を説明する。

【0018】

#### 1. 階層型負荷

本発明の実施の形態では、階層型負荷を推定するための技術が提供される。階層型負荷とは、複数の処理層を経て呼び出される処理によって生じる負荷である。例えば、処理  $a_1 \sim a_p$  のそれぞれは処理  $b_1 \sim b_q$  のいずれかを呼び出し、処理  $b_1 \sim b_q$  のそれぞれは処理  $c_1 \sim c_r$  のいずれかを呼び出すとする。このとき、処理  $c_1 \sim c_r$  によって生じる負荷は階層型負荷である。なぜなら、処理  $c_1 \sim c_r$  は、処理  $a_1 \sim a_p$  からなる処理層と処理  $b_1 \sim b_q$  からなる処理層を経て、つまり複数の処理層を経て呼び出される処理だからである。

20

【0019】

多層システムでは、上位層の処理が下位層の処理を呼び出すため、階層型負荷が発生する。図1は、多層システムにおいて階層的に呼び出される階層処理を説明するための概念図である。ユーザーが直接呼び出す処理、または、システムの最上位で呼び出される処理は、以下「サービス」と参照される。それ以外の処理は、以下「プロシージャ」と参照される。階層処理において、サービスはプロシージャを呼び出す。また、階層処理において、プロシージャが他のプロシージャを呼び出すことが許される。このとき、呼び出す側（上位層側）のプロシージャは「上位プロシージャ」と参照され、呼び出される側（下位層側）のプロシージャは「下位プロシージャ」と参照される。

30

【0020】

図1に示されるように、サービス ( $SA, SB, SC \dots$ ) の各々は、プロシージャ ( $P1A, P1B, P1C \dots$ ) のうち少なくとも1つを呼び出す。このとき、サービス ( $SA, SB, SC \dots$ ) は最上位プロシージャに相当し、プロシージャ ( $P1A, P1B, P1C \dots$ ) は下位プロシージャに相当する。更に、プロシージャ ( $P1A, P1B, P1C \dots$ ) は、プロシージャ ( $P2A, P2B, P2C \dots$ ) のうち少なくとも1つを呼び出してよい。このとき、プロシージャ ( $P1A, P1B, P1C \dots$ ) が上位プロシージャに相当し、プロシージャ ( $P2A, P2B, P2C \dots$ ) が下位プロシージャに相当する。同様にして、プロシージャ ( $PnA, PnB, PnC \dots$ ) まで呼び出され得る。ここで、 $n$  は1以上の整数である。階層処理における最下位のプロシージャ ( $PnA, PnB, PnC \dots$ ) は、以下「最下位プロシージャ」と参照される。

40

【0021】

また、階層処理のうちある階層の処理（プロシージャ）は、データベース層を操作するためのデータベース操作文を発行する。例えば、データベース操作文は、DML文である。データベース層に対してDML文等のデータベース操作文を発行する処理は、データベ

50

ース操作処理と参照される。本実施の形態において、データベース操作処理は、最下位プロシージャ（ $P_nA$  ,  $P_nB$  ,  $P_nC$  . . . ）であるとする。

#### 【0022】

階層処理の最下位プロシージャ（ $P_nA$  ,  $P_nB$  ,  $P_nC$  . . . ）は、それぞれ所定のDML文を発行する。このDML文は、多層システムのうち最下位層である「データベース層」への処理要求として発行される。最下位プロシージャは複数の処理層を経て呼び出されるため、その最下位プロシージャによって発行されるDML文の負荷は、階層型負荷である。すなわち、データベース層には、階層型負荷であるDML文の負荷がかかる。

#### 【0023】

多層システムにおいて、データベース層はボトルネック層となりがちであり、データベース層にかかる階層型負荷を見積もることが重要である。本発明の実施の形態によれば、設計フェーズにおいても、データベース層にかかる階層型負荷を定量的に推定することができる技術が提供される。そのためのシステムが、次に説明される「階層型負荷推定システム」である。

#### 【0024】

### 2. 階層型負荷推定システム

図2は、本実施の形態に係る階層型負荷推定システム1の構成例を示すブロック図である。階層型負荷推定システム1は、記憶装置2、処理装置3、入力装置4、出力装置5を備えるコンピュータシステムである。記憶装置2は、RAM（Random Access Memory）やHDD（Hard Disk Drive）等である。処理装置3は、CPU（Central Processing Unit）を有し、プログラムを実行することによって様々な機能を提供する。入力装置4は、キーボード、マウス、メディアドライブ、入力インターフェース等である。出力装置5は、ディスプレイ、メディアドライブ、出力インターフェース等である。

#### 【0025】

記憶装置2には、階層呼出情報10、DML発行情報20、DML負荷情報30、推定負荷情報RESなどが格納される。階層呼出情報10は、外部呼出回数情報11とプロシージャ呼出情報12とを含んでいる。これら階層呼出情報10、DML発行情報20、DML負荷情報30は、階層型負荷の算出に用いられるデータであり、推定負荷情報RESは、その算出結果を示す。つまり、階層型負荷推定システム1は、階層呼出情報10、DML発行情報20、DML負荷情報30を入力とし、推定負荷情報RESを出力とする。

#### 【0026】

階層呼出情報10は、既出の図1で示されるような階層処理の設計を示す。より詳細には、階層呼出情報10は、外部呼出回数情報11とプロシージャ呼出情報12とを含んでいる。

#### 【0027】

外部呼出回数情報11は、外部から要求されるサービスに関連する情報であり、サービスの想定される呼出回数を示す。図3は、外部呼出回数情報11の一例を示している。図3に示されるように、外部呼出回数情報11は、複数のサービス（ $SA$  ,  $SB$  ,  $SC$  . . . ）のそれぞれに関して、サービス名と想定される呼出回数との対応関係を示している。

#### 【0028】

一方、プロシージャ呼出情報12は、階層処理における上位プロシージャと下位プロシージャとの間の呼び出し関係を規定する情報である。既出の図1に示されるように、階層処理の設計に応じて、プロシージャ呼出情報12は、 $n$ 種類のプロシージャ呼出情報12-1 ~ 12-nを含み得る（ $n$ は1以上の整数）。プロシージャ呼出情報12-1は、サービス（ $SA$  ,  $SB$  ,  $SC$  . . . ）と下位プロシージャ（ $P_1A$  ,  $P_1B$  ,  $P_1C$  . . . ）との間の呼び出し関係を規定する。プロシージャ呼出情報12-2は、上位プロシージャ（ $P_1A$  ,  $P_1B$  ,  $P_1C$  . . . ）と下位プロシージャ（ $P_2A$  ,  $P_2B$  ,  $P_2C$  . . . ）との間の呼び出し関係を規定する。プロシージャ呼出情報12-nは、上位プロシージャと最下位プロシージャ（ $P_nA$  ,  $P_nB$  ,  $P_nC$  . . . ）との間の呼び出し関係を規定する。

10

20

30

40

50

## 【 0 0 2 9 】

図 4 A は、プロシージャ呼出情報 1 2 の一例を示している。例えばプロシージャ呼出情報 1 2 - 1 は、「サービス S A がプロシージャ P 1 B を呼び出すこと」、「サービス S C がプロシージャ P 1 A 及び P 1 C を呼び出すこと」などを示している。プロシージャ呼出情報 1 2 - 2 は、「プロシージャ P 1 A がプロシージャ P 2 A 及び P 2 B を呼び出すこと」などを示している。

## 【 0 0 3 0 】

図 4 B は、プロシージャ呼出情報 1 2 の他の例を示している。プロシージャ呼出情報 1 2 は、上位プロシージャが下位プロシージャを呼び出す回数を示していてもよい。例えば図 4 B において、プロシージャ呼出情報 1 2 - 2 は、「プロシージャ P 1 A が、プロシージャ P 2 A を 2 回呼び出し、プロシージャ P 2 B を 1 回呼び出すこと」などを示している。

10

## 【 0 0 3 1 】

再度図 1 を参照して、DML 発行情報 2 0 は、最下位プロシージャ ( P n A , P n B , P n C . . . ) によって発行される DML 文を規定する情報である。例えば、DML 発行情報 2 0 は、各エンティティに対して発行される DML 文の発行回数を、最下位プロシージャ ( P n A , P n B , P n C . . . ) 毎に示す。ここで、エンティティとは、データベース中のアクセス対象のテーブル ( データセット ) を意味する。

## 【 0 0 3 2 】

また、本実施の形態において、DML 文は、様々なパラメータに基づいて分類されてもよい。その様々なパラメータとしては、DML 文の種類、CPU 使用時間、エンティティ、データアクセス量、実行計画等が挙げられる。DML 文が分類される場合、DML 発行情報 2 0 は、発行される DML 文の分類を、最下位プロシージャ ( P n A , P n B , P n C . . . ) 毎に示す。例えば、DML 発行情報 2 0 は、各エンティティに対して発行される DML 文の「種類」を、最下位プロシージャ毎に示す。DML 文の種類としては、データ抽出を指示する “ s e l e c t 文 ”、データ更新を指示する “ u p d a t e 文 ”、データ挿入を指示する “ i n s e r t 文 ”、データ削除を指示する “ d e l e t e 文 ” などが挙げられる。

20

## 【 0 0 3 3 】

図 5 は、DML 発行情報 2 0 の一例を示している。図 5 で示される例では、DML 文は、種類とエンティティに基づいて分類されている。尚、“ C ” は “ i n s e r t ” を、“ R ” は “ s e l e c t ” を、“ U ” は “ u p d a t e ” を、“ D ” は “ d e l e t e ” をそれぞれ表し、このような図は一般に C R U D 図と呼ばれている。この DML 発行情報 2 0 は、最下位プロシージャ ( P n A , P n B , P n C . . . ) の各々が、どのエンティティに対してどのような種類の DML 文を発行するかを示している。例えば、最下位プロシージャ P n B は、エンティティ E 1 に対して s e l e c t 文を発行し、エンティティ E 3 に対して s e l e c t 文及び u p d a t e 文を発行する。このような DML 発行情報 2 0 に対して、更に各分類の DML 文の発行回数が付与されていてもよい。

30

## 【 0 0 3 4 】

DML 負荷情報 3 0 は、DML 文あたりの負荷を示す情報である。負荷としては、CPU 使用時間、CPU 使用時間を正規化することにより得られる CPU 使用率、ロック競合、容量 ( テーブル等が使用する記憶領域の大きさ ) などが挙げられる。DML 文が分類される場合、DML 負荷情報 3 0 は、DML 文あたりの負荷を、DML 文の分類毎に示す。例えば、DML 負荷情報 3 0 は、各エンティティに対して発行される DML 文あたりの負荷を、DML 文の種類毎に示す。

40

## 【 0 0 3 5 】

図 6 は、DML 負荷情報 3 0 の一例を示している。図 6 では、各エンティティ ( E 1 、 E 2 、 E 3 ) に対して発行される DML 文あたりの負荷が、DML 文の種類 ( s e l e c t 、 u p d a t e 、 i n s e r t 、 d e l e t e ) 毎に示されている。負荷は、CPU 使用率である。例えば、エンティティ E 1 に対して発行される s e l e c t 文の CPU 使用

50



率は10であり、update文のCPU使用率は103である。同じエンティティに対するDML文であっても、DML文の種類に依存して負荷が異なることに留意されたい。

【0036】

以上に説明された外部呼出回数情報11、プロシージャ呼出情報12、DML発行情報20、及びDML負荷情報30は、設計フェーズにおいて得ることができる設計情報であることに留意されたい。

【0037】

再度図2を参照して、記憶装置2には更に、階層型負荷推定プログラムPROGが格納される。階層型負荷推定プログラムPROGは、階層型負荷推定システム1に階層型負荷の算出処理を実行させるコンピュータプログラムである。典型的には、階層型負荷推定プログラムPROGは、コンピュータ読み取り可能な記録媒体に記録され、その記録媒体から記憶装置2に読み込まれる。

10

【0038】

処理装置3は、階層型負荷推定プログラムPROGを実行することにより、階層型負荷の算出処理を行う。具体的には、処理装置3は、呼出回数算出モジュール100、発行回数算出モジュール200、及び負荷算出モジュール300を有している。これら呼出回数算出モジュール100、発行回数算出モジュール200、及び負荷算出モジュール300は、処理装置3が階層型負荷推定プログラムPROGを実行することにより実現され、それぞれが所定のデータ処理機能を提供する。以下、呼出回数算出モジュール100、発行回数算出モジュール200、及び負荷算出モジュール300によるデータ処理を詳細に説明する。

20

【0039】

3. 処理フロー

図7及び図8のそれぞれは、階層型負荷システム1による処理フローを示すブロック図及びフローチャートである。

【0040】

ステップS100:

呼出回数算出モジュール100は、記憶装置2から階層呼出情報10(外部呼出回数情報11及びプロシージャ呼出情報12)を読み出す。そして、呼出回数算出モジュール100は、階層呼出情報10を参照して、最下位プロシージャ(PnA, PnB, PnC・・・)の呼出回数を算出する。上述の通り、外部呼出回数情報11は、最上位のサービス(SA, SB, SC・・・)のそれぞれの呼出回数を示しており、プロシージャ呼出情報12は、上位プロシージャと下位プロシージャとの間の呼び出し関係を規定している。従って、呼出回数算出モジュール100は、外部呼出回数情報11及びプロシージャ呼出情報12を用いることによって、各層のプロシージャの呼出回数を上位層から下位層に向けて順番に算出することができる。結果として、最下位プロシージャ(PnA, PnB, PnC・・・)の呼出回数が算出される。算出された呼出回数は、最下位呼出回数情報15として記憶装置2に格納される。

30

【0041】

ステップS200:

発行回数算出モジュール200は、記憶装置2から最下位呼出回数情報15とDML発行情報20を読み出す。上述の通り、最下位プロシージャ(PnA, PnB, PnC・・・)に関して、最下位呼出回数情報15は呼出回数を示し、DML発行情報20は発行されるDML文を示している。従って、発行回数算出モジュール200は、最下位呼出回数情報15とDML発行情報20を参照することにより、最下位プロシージャによって発行されるDML文の発行回数を算出することができる。つまり、発行回数算出モジュール200は、階層処理において発行されるDML文の発行回数を算出する。DML文が上述のように分類される場合、DML文の発行回数は分類毎に算出されてもよい。算出されたDML文の発行回数は、発行回数情報25として記憶装置2に格納される。

40

【0042】

50

**ステップ S 3 0 0 :**

負荷算出モジュール 3 0 0 は、記憶装置 2 から発行回数情報 2 5 と D M L 負荷情報 3 0 を読み出す。上述の通り、発行回数情報 2 5 は D M L 文の発行回数を示し、D M L 負荷情報 3 0 は D M L 文あたりの負荷を示している。従って、負荷算出モジュール 3 0 0 は、発行回数情報 2 5 と D M L 負荷情報 3 0 を参照することにより、最下位プロシージャによって発行される D M L 文の負荷を算出することができる。つまり、負荷算出モジュール 3 0 0 は、階層処理において発行される D M L 文の負荷を算出する。D M L 文が上述のように分類される場合、D M L 文の負荷は分類毎に算出されてもよい。算出された D M L 文の負荷は、推定負荷情報 R E S として記憶装置 2 に格納される。算出された D M L 文の負荷は、多層システムのデータベース層にかかる負荷の推定値（推定負荷）を表している。つまり、推定負荷情報 R E S は、データベース層にかかる階層型負荷の推定値を示している。システム構築者は、推定負荷情報 R E S を参照することによって、十分なシステム性能を達成するために必要な計算機資源を見積もることができる。

10

**【 0 0 4 3 】**

以下、ステップ S 1 0 0 とステップ S 2 0 0 を、更に詳しく説明する。

**【 0 0 4 4 】****3 - 1 . ステップ S 1 0 0**

ステップ S 1 0 0 において、呼出回数算出モジュール 1 0 0 は、階層呼出情報 1 0 から最下位呼出回数情報 1 5 を作成する。図 9 は、ステップ S 1 0 0 を示すフローチャートである。

20

**【 0 0 4 5 】**

まず、ループ処理に用いられるカウンタ  $i$  が 1 に設定される（ステップ S 1 1 0）。このループ処理は、プロシージャ呼出情報 1 2 - 1 ~ 1 2 -  $n$  の数だけ実施される。つまり、カウンタ  $i$  は初期値 1 から最終値  $n$  まで 1 ずつ変化する。

**【 0 0 4 6 】**

各ループ処理において、記憶装置 2 からプロシージャ呼出情報 1 2 -  $i$  が読み込まれる（ステップ S 1 2 0）。また、上位呼出回数情報が読み込まれる（ステップ S 1 3 0）。ここで、「上位呼出回数情報」とは、プロシージャ呼出情報 1 2 -  $i$  中の上位プロシージャのそれぞれの呼出回数を示す情報である。次に、その上位プロシージャの呼出回数とプロシージャ呼出情報 1 2 -  $i$  に基づいて、プロシージャ呼出情報 1 2 -  $i$  中の下位プロシージャのそれぞれの呼出回数が算出される（ステップ S 1 4 0）。下位プロシージャのそれぞれの呼出回数を示す情報が、「下位呼出回数情報」である。カウンタ  $i$  が最終値  $n$  に満たない場合（ステップ S 1 6 0 ; N o）、カウンタ  $i$  に 1 が加算され（ステップ S 1 7 0）、次のループ処理が実施される。

30

**【 0 0 4 7 】**

具体的には、最初カウンタ  $i$  は 1 であるため、プロシージャ呼出情報 1 2 - 1 が参照される。このとき、上位プロシージャはサービス（S A , S B , S C . . .）であり、上位呼出回数情報は外部呼出回数情報 1 1（図 3 参照）である。従って、記憶装置 2 から外部呼出回数情報 1 1 が読み出される。外部呼出回数情報 1 1 は、サービス（S A , S B , S C . . .）のそれぞれの呼出回数を示しており、プロシージャ呼出情報 1 2 - 1 はそれらサービスとプロシージャ（P 1 A , P 1 B , P 1 C . . .）との間の呼び出し関係を規定している。従って、外部呼出回数情報 1 1 とプロシージャ呼出情報 1 2 - 1 から、下位プロシージャ（P 1 A , P 1 B , P 1 C . . .）のそれぞれの呼出回数を示す下位呼出回数情報を作成することができる。

40

**【 0 0 4 8 】**

次に、カウンタ  $i$  は 2 となり、プロシージャ呼出情報 1 2 - 2 が参照される。このとき、上位プロシージャはプロシージャ（P 1 A , P 1 B , P 1 C . . .）である。従って、前回のループ処理で作成された下位呼出回数情報が、今回のループ処理における上位呼出回数情報として利用される。そして、プロシージャ呼出情報 1 2 - 2 と上位呼出回数情報に基づいて、下位プロシージャ（P 2 A , P 2 B , P 2 C . . .）のそれぞれの呼出回数

50

が算出される。

【 0 0 4 9 】

カウンタ  $i$  が最終値  $n$  に達するまで、同様のループ処理が繰り返される。最終的には、最下位プロシージャ (  $P_n A$  ,  $P_n B$  ,  $P_n C$  . . . ) のそれぞれの呼出回数が算出される。カウンタ  $i$  が最終値  $n$  の場合 ( ステップ  $S 1 6 0$  ;  $Y e s$  )、ループ処理は終了する。最終的に得られた最下位プロシージャの呼出回数を示す情報が、最下位呼出回数情報 1 5 である ( ステップ  $S 1 8 0$  )。このように、本実施の形態によれば、各層のプロシージャの呼出回数が、上位層から下位層に向けて順番に算出される。結果として、最下位プロシージャの呼出回数が算出され、最下位呼出回数情報 1 5 が作成される。

【 0 0 5 0 】

上記ステップ  $S 1 4 0$  において下位プロシージャの呼出回数を算出する方法としては、様々な方法が考えられる。図 1 0 は、その一例を示すフローチャートである。まず、下位エントリが作成される ( ステップ  $S 1 4 1$  )。下位エントリとは、下位プロシージャの識別子と呼出回数のフィールドからなるテーブルである。下位エントリの各フィールド値を算出することがステップ  $S 1 4 0$  の目的である。各フィールドの初期値は全て 0 に設定される。

【 0 0 5 1 】

次に、上位プロシージャのリストを示す上位リストが作成される ( ステップ  $S 1 4 2$  )。続いて、上位リストから任意の 1 つの上位プロシージャが選択される ( ステップ  $S 1 4 3$  )。次に、選択上位プロシージャによって呼び出される下位プロシージャのそれぞれの呼出回数が算出される ( ステップ  $S 1 5 0$  )。算出された下位プロシージャの呼出回数は、下位エントリ中の対応するフィールド値に加算される。その後、選択上位プロシージャが上位リストから削除される ( ステップ  $S 1 4 4$  )。上位リストに他の上位プロシージャが残っていれば ( ステップ  $S 1 4 5$  ;  $Y e s$  )、処理はステップ  $S 1 4 3$  に戻る。上位プロシージャが残っていなければ ( ステップ  $S 1 4 5$  ;  $N o$  )、ステップ  $S 1 4 0$  は終了する。

【 0 0 5 2 】

ステップ  $S 1 5 0$  は、次の通りである。まず、選択上位プロシージャによって呼び出される下位プロシージャのリストを示す下位リストが作成される ( ステップ  $S 1 5 1$  )。続いて、下位リストから任意の 1 つの下位プロシージャが選択される ( ステップ  $S 1 5 2$  )。次に、選択下位プロシージャの呼出回数が算出される。このとき、選択上位プロシージャの総呼出回数は上位呼出回数情報によって与えられている。よって、プロシージャ呼出情報 1 2 -  $i$  が上位プロシージャによって呼び出される下位プロシージャの種類を示している場合 ( 図 4 A 参照 )、選択上位プロシージャの総呼出回数が選択下位プロシージャの呼出回数となる。また、プロシージャ呼出情報 1 2 -  $i$  が 1 回の選択上位プロシージャによって呼び出される選択下位プロシージャの回数を示している場合 ( 図 4 B 参照 )、その回数に選択上位プロシージャの総呼出回数を掛けることによって、選択下位プロシージャの呼出回数が算出される。算出された選択下位プロシージャの呼出回数は、下位エントリ中の対応するフィールド値に加算され、下位エントリが更新される ( ステップ  $S 1 5 3$  )。その後、選択下位プロシージャが下位リストから削除される ( ステップ  $S 1 5 4$  )。下位リストに他の下位プロシージャが残っていれば ( ステップ  $S 1 5 5$  ;  $Y e s$  )、処理はステップ  $S 1 5 2$  に戻る。下位プロシージャが残っていなければ ( ステップ  $S 1 5 5$  ;  $N o$  )、ステップ  $S 1 5 0$  は終了する。

【 0 0 5 3 】

3 - 2 . ステップ  $S 2 0 0$

ステップ  $S 2 0 0$  において、発行回数算出モジュール 2 0 0 は、最下位呼出回数情報 1 5 及び  $D M L$  発行情報 2 0 から発行回数情報 2 5 を算出する。図 1 1 は、ステップ  $S 2 0 0$  を示すフローチャートである。まず、最下位呼出回数情報 1 5 が記憶装置 2 から読み込まれる ( ステップ  $S 2 1 0$  )。また、 $D M L$  発行情報 2 0 が記憶装置 2 から読み込まれる ( ステップ  $S 2 2 0$  )。

10

20

30

40

50

## 【 0 0 5 4 】

次に、エンティティエントリが作成される（ステップ S 2 3 0）。エンティティエントリとは、DML 発行情報 2 0 で示されるエンティティの識別子と DML 文の発行回数のフィールドからなるテーブルである。上述のように DML 文が分類される場合、発行回数のフィールドは、DML 文の分類毎に設けられる。このエンティティエントリの各フィールド値を算出することがステップ S 2 0 0 の目的である。各フィールドの初期値は全て 0 に設定される。

## 【 0 0 5 5 】

次に、最下位リストが作成される（ステップ S 2 4 0）。最下位リストは、最下位呼出回数情報 1 5 で示される最下位プロシージャ（P n A, P n B, P n C・・・）のリストである。続いて、最下位リストから任意の 1 つの最下位プロシージャが選択される（ステップ S 2 5 0）。

10

## 【 0 0 5 6 】

次に、選択された最下位プロシージャによって発行される DML 文の発行回数が算出される。上述の通り、最下位呼出回数情報 1 5 は、各最下位プロシージャの総呼出回数を示しており、DML 発行情報 2 0 は、各最下位プロシージャによって発行される DML 文を規定している。DML 発行情報 2 0 が、1 回の最下位プロシージャによる DML 文の発行回数を示している場合を考える。この場合、その発行回数に選択最下位プロシージャの総呼出回数を掛けることによって、選択最下位プロシージャによる DML 文の発行回数の合計が算出される。DML 文が分類されている場合には、その分類毎に、DML 文の発行回数が同様に算出される。このようにして算出された DML 文の発行回数は、エンティティエントリ中の対応するフィールド値に加算され、エンティティエントリが更新される（ステップ S 2 6 0）。

20

## 【 0 0 5 7 】

その後、選択最下位プロシージャが最下位リストから削除される（ステップ S 2 7 0）。最下位リストに他の最下位プロシージャが残っていれば（ステップ S 2 8 0 ; Yes）、処理はステップ S 2 5 0 に戻る。最下位プロシージャが残っていなければ（ステップ S 2 8 0 ; No）、DML 文の発行回数の算出処理は終了する。最終的なエンティティエントリが、発行回数情報 2 5 に相当する。つまり、階層処理において発行される DML 文の発行回数を示す発行回数情報 2 5 が決定される（ステップ S 2 9 0）。

30

## 【 0 0 5 8 】

## 4 . 処理例

次に、本実施の形態に係る階層型負荷推定システム 1 による具体的な処理例を説明する。

## 【 0 0 5 9 】

## 4 - 1 . 第 1 の処理例

図 1 2 は、第 1 の処理例における外部呼出回数情報 1 1 を示している。外部呼出回数情報 1 1 は、3 種類のサービス S A、S B、S C のそれぞれの呼出回数を示している。ここで、各呼出回数は、1 秒あたりに想定されるサービス要求回数である。尚、単位時間は、1 秒に限られず、任意の値に設定され得る。

40

## 【 0 0 6 0 】

図 1 3 は、第 1 の処理例におけるプロシージャ呼出情報 1 2 を示している。簡単のため、パラメータ n は 1 であるとする。つまり、最下位プロシージャは、サービス S A ~ S C によって呼び出されるプロシージャ P 1 A ~ P 1 D であるとする。パラメータ n が 2 以上であっても、同様の議論が適用される。図 1 3 に示されるように、プロシージャ呼出情報 1 2 - 1 は、サービス S A ~ S C とプロシージャ P 1 A ~ P 1 D との間の呼び出し関係を示している。具体的には、サービス S A はプロシージャ P 1 A、P 1 B を呼び出し、サービス S B はプロシージャ P 1 B、P 1 C を呼び出し、サービス S C はプロシージャ P 1 C、P 1 D を呼び出す。

## 【 0 0 6 1 】

50

図14は、第1の処理例におけるDML発行情報20を示している。第1の処理例では、簡単のため、データベース層が1つのエンティティとして扱われる。図14に示されるように、DML発行情報20は、そのエンティティに対するDML文の発行回数を、最下位プロシージャP1A～P1D毎に示している。具体的には、最下位プロシージャP1A～P1Dの各々は、1回の呼び出しあたり2つのDML文を発行する。

#### 【0062】

図15は、第1の処理例におけるDML負荷情報30を示している。DML負荷情報30は、DML発行情報20の形式に適合するように作成されている。つまり、データベース層が1つのエンティティとして扱われる。図15に示されるように、DML負荷情報30は、そのエンティティに対する1つのDML文あたりの負荷を示している。ここで、負荷は、平均実行時間である。図15で示される例では、全てのDML文の平均実行時間は6msである。

#### 【0063】

ステップS100において、図12で示された外部呼出情報11と図13で示されたプロシージャ呼出情報12に基づいて、最下位プロシージャP1A～P1Dのそれぞれの呼出回数が算出される。ステップS140(図10参照)で使用される下位エントリ、上位リスト、及び下位リストは、プロシージャ呼出情報12から容易に作成される。すなわち、下位エントリは、最下位プロシージャP1A～P1Dのそれぞれの呼出回数のフィールドを有する。上位リストは、サービスSA、SB、SCを含む。サービスSAに対応する下位リストは、最下位プロシージャP1A、P1Bを含む。サービスSBに対応する下位リストは、最下位プロシージャP1B、P1Cを含む。サービスSCに対応する下位リストは、最下位プロシージャP1C、P1Dを含む。ステップS100の結果、図16に示される最下位呼出回数情報15が得られる。最下位プロシージャP1A、P1B、P1C、P1Dのそれぞれの呼出回数は、それぞれ、3、13、35、25である。

#### 【0064】

ステップS200において、図16で示された最下位呼出回数情報15と図14で示されたDML発行情報20に基づいて、DML文の発行回数の合計が算出される。エンティティエントリは、DML発行情報20から容易に作成される。第1の例では、データベース層が1つのエンティティとして扱われるため、エンティティエントリに含まれるエンティティの識別子も1つだけである。ステップS200の結果、図17に示される発行回数情報25が得られる。最下位プロシージャP1A～P1Dの各々は、1回の呼び出しあたり2つのDML文を発行するため、DML文の発行回数の合計は152回である。

#### 【0065】

ステップS300において、図17で示された発行回数情報25と図15で示されたDML負荷情報30に基づいて、DML文の負荷が算出される。DML文の発行回数は合計152回であり、DML文あたりの平均実行時間は6msであるため、DML文の総実行時間は912ms(=152×6ms)と算出される。ここで、これらDML文の発行は、1秒あたりに想定されるサービス要求(図12参照)の結果であることに留意されたい。つまり、1秒あたり912msの期間が、サービス要求の結果としてのDML文の実行に費やされる。言い換えれば、CPU使用率は91.2%である。このCPU使用率=91.2%が、本例によって推定される負荷である。図18は、本例において作成される推定負荷情報RESを示している。

#### 【0066】

尚、DML文の発行回数(=152回)も負荷に関連するパラメータである。よって、推定負荷情報RESには、ステップS300で算出される負荷だけでなく、ステップS200で算出されるDML文の発行回数が含まれていてもよい。あるいは、推定負荷情報RESとして、ステップS200で算出されるDML文の発行回数だけが与えられてもよい。

#### 【0067】

推定されたCPU使用率(91.2%)を利用して、システム構築者は、十分なシステ

10

20

30

40

50

ム性能を達成するために必要な計算機資源を、設計フェーズにおいても見積もることができる。例えば、CPU使用率が92%以下であれば運用上問題ない場合、CPUは1つで十分であることが推定される。CPU使用率を50%以下に保つ必要がある場合、システム構築者は、CPU数を2つに増やすことを検討することができる。あるいは、システム構築者は、DML文とデータ構造の最適化をより高いレベルで実現する必要があることを推定できる。あるいは、システム構築者は、提供するサービスの量を抑制することを検討することができる。

#### 【0068】

また、複数種類の外部呼出回数情報11に対して、つまり、異なるサービス提供シナリオに対して、それぞれ負荷を推定することもできる。この場合、一部または全てのサービス提供シナリオを採用するのに十分な計算機資源の量を推定することができる。または、想定される計算機資源の範囲内で適用可能なサービス提供シナリオとそれ以外を選別することができる。このように、本実施の形態は、設計フェーズにおけるサービス提供シナリオの策定を支援することができる。

10

#### 【0069】

本実施の形態に係る負荷推定は、新規にシステムを構築する場合だけでなく、既存システムが変更される場合にも実施され得る。例えば、既存システムのDML発行情報20とプロシージャ呼出情報12に変更が反映された後、負荷が推定される。また、サービス提供シナリオが変更される場合、外部呼出回数情報11に変更が反映された後、負荷が推定される。また、システムを構成するハードウェアが変更される場合（例えば、CPUが置き換えられる場合）、DML負荷情報30にハードウェアスペックの変更が反映された後、負荷が推定される。

20

#### 【0070】

##### 4-2. 第2の処理例

第2の処理例において、外部呼出回数情報11とプロシージャ呼出情報12は第1の処理例と同じである。従って、ステップS100の結果は第1の処理例と同じであり、最下位呼出回数情報15は、既出の図16で示されたようになる。

#### 【0071】

図19は、第2の処理例におけるDML発行情報20を示している。第2の処理例では、DML文は分類され、DML発行情報20は、発行されるDML文の分類を最下位プロシージャP1A~P1D毎に示している。より詳細には、DML文は、「DML文の種類」と「エンティティ」に基づいて分類される。ここでは、4つのエンティティE1~E4と、4つの種類“C(insert)”、“R(select)”、“U(update)”、“D(delete)”が考慮される。最下位プロシージャP1A~P1Dの各々は、エンティティE1~E4の少なくともいずれかに対して、いずれかの種類のDML文を発行する。図19に示されるように、DML発行情報20は、最下位プロシージャP1A~P1Dのそれぞれによって発行されるDML文のエンティティと種類を示している。例えば、1回の最下位プロシージャP1Aは、エンティティE1に対してselect文を発行し、また、エンティティE4に対してdelete文を発行する。

30

#### 【0072】

図20は、第2の処理例におけるDML負荷情報30を示している。DML負荷情報30は、DML発行情報20の形式に適合するように作成されている。つまり、DML負荷情報30は、DML文あたりの負荷を、DML文の分類毎に示している。第1の処理例と同様に、負荷は平均実行時間である。例えば、エンティティE1に対するselect文の平均実行時間は2msであり、エンティティE3に対するupdate文の平均実行時間は14msである。尚、空白のフィールドが残されているのは、発行されないDML文の実行時間は不要であるためである。

40

#### 【0073】

ステップS200において、図16で示された最下位呼出回数情報15と図19で示されたDML発行情報20に基づいて、DML文の発行回数が算出される。このとき、DM

50

L文の発行回数は、上記DML文の分類毎に算出される。従って、エンティティエントリ（発行回数情報25）は、DML文の発行回数を分類毎に示すテーブルとなる。そのようなテーブル形式は、DML負荷情報30において負荷（実行時間）が発行回数で置き換えられたものと同じである。ステップS200の結果、図21に示される発行回数情報25が得られる。例えば、エンティティE1に対してselect文が合計28回発行され、エンティティE3に対してupdate文が合計35回発行される。

【0074】

ステップS300において、図21で示された発行回数情報25と図20で示されたDML負荷情報30に基づいて、DML文の負荷が算出される。このとき、DML文の負荷は、上記DML文の分類毎に算出される。ステップS300の結果、図22に示される推定負荷情報RESが得られる。図22に示される推定負荷情報RESは、1秒あたりのDML文の実行時間[ms]、すなわち、1秒あたりのCPU使用時間[ms]を、DML文の分類毎に示している。尚、1秒あたりのCPU使用時間[ms]を10で割れば、CPU使用率[%]が得られる。また、図22で示される推定負荷情報RESには、「total行」と「total列」が付加されている。total行には種類毎のCPU使用時間の部分和が示されている。一方、total列にはエンティティ毎のCPU使用時間の部分和が示されている。total行とtotal列の双方に属するフィールドには、全DML文のCPU使用時間の総和（851ms）が示されている。

【0075】

推定負荷情報RESには、ステップS200で算出されたDML文の分類毎の発行回数が含まれていてもよい。第2の処理例で作成された推定負荷情報RESは、第1の処理例と同様に利用できる。更に、DML文の負荷や発行回数が分類毎に算出されているため、次のような利用方法も可能である。

【0076】

分類毎に算出された負荷は、設計段階において特に注意を払うべきDML文の特定に有効である。例えば、図22で示された推定負荷情報RESから、エンティティE3に対するupdate文の負荷（490ms）が総負荷（851ms）の半分以上を占めていることがわかる。そのため、システム構築者は、性能を向上させるためにはエンティティE3に対するupdate文を最適化することが効率的であると判断することができる。

【0077】

また、エンティティ毎の負荷の部分和は、設計段階において特に注意を払うべきエンティティの特定に有効である。例えば、図22で示された推定負荷情報RESから、エンティティE3に関する負荷（560ms）が総負荷（851ms）の半分以上を占めていることが分かる。そのため、システム構築者は、性能を向上させるためにはエンティティE3の設計を最適化することが効率的であると判断することができる。一方、エンティティE2に関する負荷の総負荷への寄与は小さいため、システム構築者は、エンティティE2の設計を最適化することによるメリットは小さいと判断できる。

【0078】

また、DML文の種類及びエンティティ毎に算出された発行回数を分析することにより、ロック競合の可能性を定量的に推定できる。基本的に、異なるエンティティにアクセスするDML文は、同じデータにアクセスすることはない。一方で、同一のエンティティにアクセスするDML文は同じデータにアクセスするため、その場合、ロック競合が発生する可能性がある。ロック競合はDML文のCPU使用時間を増加させるとともに遅延を増大させるため、設計フェーズの段階から考慮すべき問題である。例えば既出の図21で示される例の場合、エンティティE1、E2に対しては、select文によるデータ読み出しのみが発生する。従って、それらエンティティE1、E2に関するロック競合が起きる可能性は極めて低いと考えられる。一方、エンティティE4に対しては、insert文とdelete文によるデータ書き込みも発生するため、ロック競合が起きる可能性がある。更に、エンティティE3に関しては、多数のupdate文によるデータ書き込みと多数のselect文によるデータ読み出しが発生するため、ロック競合が起きる可能

10

20

30

40

50

性はエンティティE 4よりも更に高くなると考えられる。このように、設計フェーズからロック競合の可能性を定量的に評価することにより、エンティティの分割等の対応を検討することが可能となる。

【0079】

また、算出された発行回数をDML文の種類別に分析することは、設計フェーズにおけるデータベースの容量設計に有効である。例えば図21で示される発行回数情報25から、エンティティE 4に対しては1秒間に、insert文が13回、delete文が3回発行されることが分かる。insert文が1行挿入を実施し、delete文が1行削除を実施するとすれば、エンティティE 4の全行数は1秒毎に10行ずつ増えると推定される。1行が64バイトの場合、エンティティE 4が必要とする記憶容量は、1秒毎に640バイト、1日に5Mバイト以上増えると推定される。このような定量的な推定は、データベースの容量設計に際して有効である。

10

【0080】

また、DML文が分類される場合、複数の異なる外部呼出回数情報11に対する負荷推定の精度を高めることができる。分類別にDML文の実行時間が異なる場合、DML文の比率が異なればDML文の平均実行時間も異なる。比率別に平均実行時間を求めるのは一般に困難なので、異なるDML文の比率、つまり異なる外部呼出回数情報11に対する負荷推定精度は低くなってしまふ。よって、分類別の平均実行時間を使えば、負荷推定精度を高めることができる。

【0081】

20

5. 効果

本発明の実施の形態によれば、少なくとも次の効果が得られる。つまり、想定されるサービス要求に対して、階層処理において発行されるDML文の数が定量的に算出される。DML文の発行回数はデータベース層にかかる負荷に関連しているため、算出されたDML文の発行回数に基づいて、データベース層にかかる負荷を見積もることができる。つまり、多層システムにおける階層型負荷を定量的に推定することができる。

【0082】

その階層型負荷の推定において、実際にシステムを構築することは不要である。たとえ設計フェーズの段階であっても、階層型負荷を定量的に推定することが可能となる。システム構築者は、推定された階層型負荷を参照することによって、十分なシステム性能を達成するために必要な計算機資源を見積もることができる。つまり、設計フェーズにおいて、想定されるサービス要求に対する階層型負荷を定量的に推定し、必要な計算機資源を見積もることができる。その結果、運用フェーズから設計フェーズへの戻りが防止され、システム構築コストが削減される。

30

【0083】

以上、本発明の実施の形態が添付の図面を参照することにより説明された。但し、本発明は、上述の実施の形態に限定されず、要旨を逸脱しない範囲で当業者により適宜変更され得る。

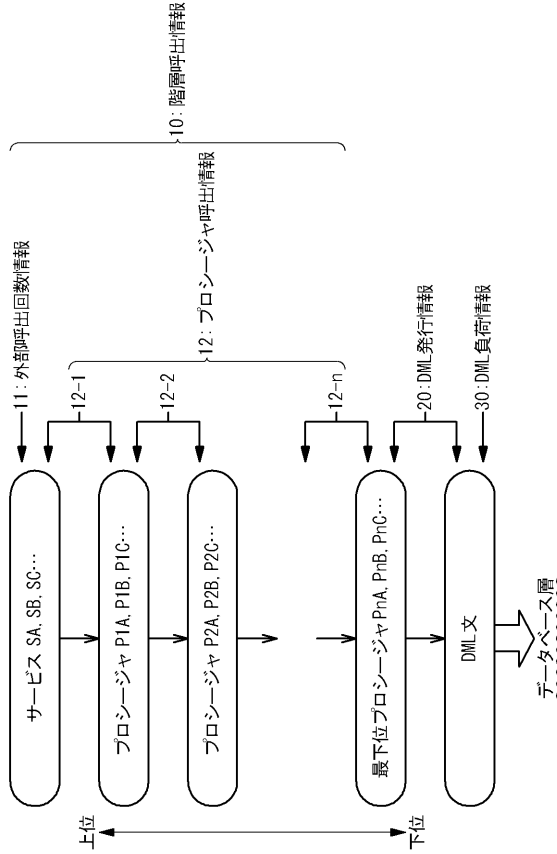
【0084】

本出願は、2007年10月3日に提出された日本国特許出願2007-259947を基礎とする優先権を主張し、その開示の全てをここに取り込む。

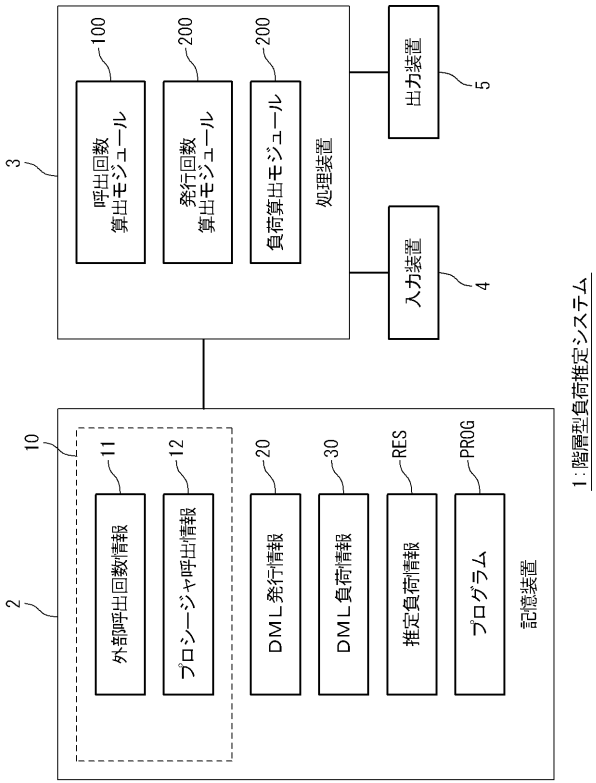
40



【図 1】



【図 2】



1: 随機型負荷推定システム

【図 3】

外部呼出回数情報 11

外部処理名	呼出回数
サービス SA	9780
サービス SB	584
サービス SC	49078
...	...

【図 4 A】

プロシージャ呼出情報 12

	サービス SA	サービス SB	サービス SC	...
12-1				
プロシージャ P1A		○	○	
プロシージャ P1B	○			
プロシージャ P1C			○	
...				
12-2				
プロシージャ P2A	○			
プロシージャ P2B	○		○	
プロシージャ P2C		○	○	
...				
12-n (n ≥ 1)				

【図4B】

プロセス呼び出し情報 12

	プロセス P1A	プロセス P1B	プロセス P1C	...
プロセス P2A	2			
プロセス P2B	1		3	
プロセス P2C		1	2	
...				

【図6】

DML負荷情報 30

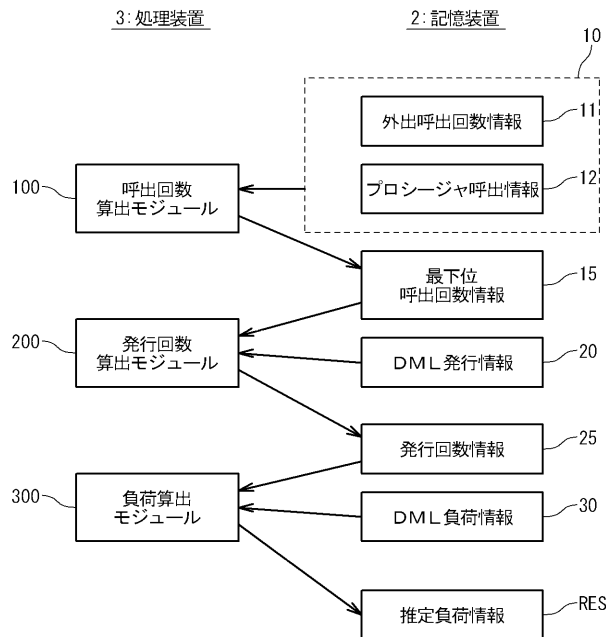
	select	update	insert	delete
エンティティ E1	10	103	50	60
エンティティ E2	15	90	41	73
エンティティ E3	12	148	67	84
...				

【図5】

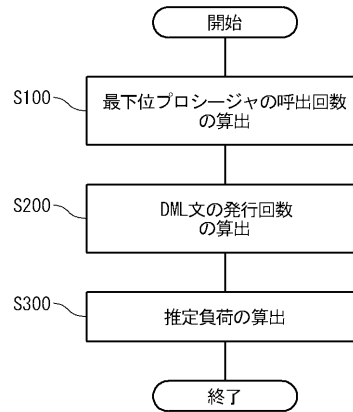
DML発行情報 20

	プロセス PnA	プロセス PnB	プロセス PnC	...
エンティティ E1	R	R	RCU	
エンティティ E2	R		RCU	
エンティティ E3		RU		
...				

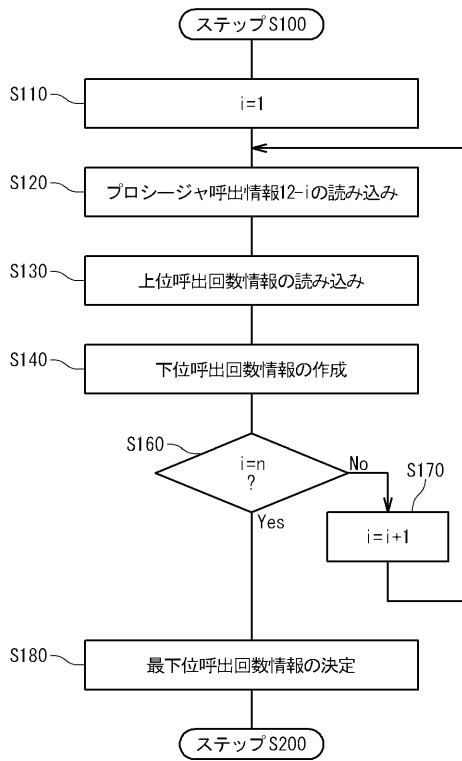
【図7】



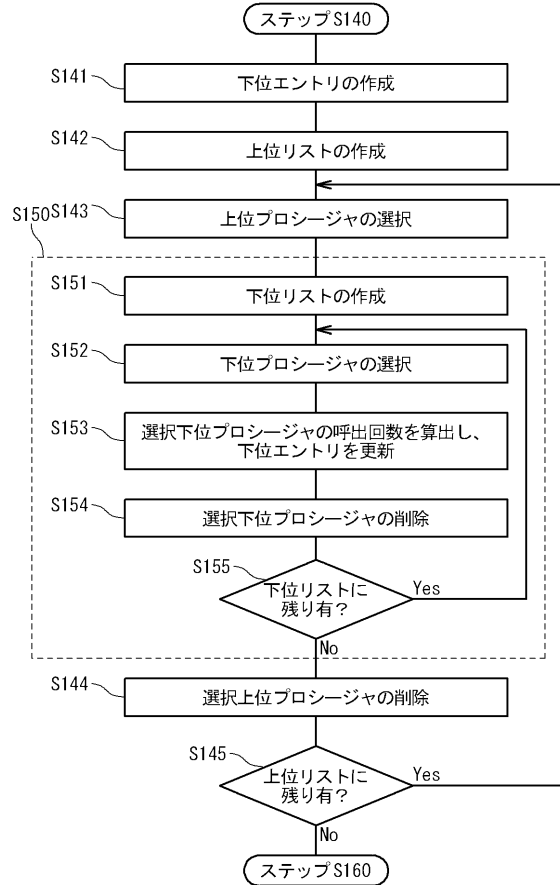
【図8】



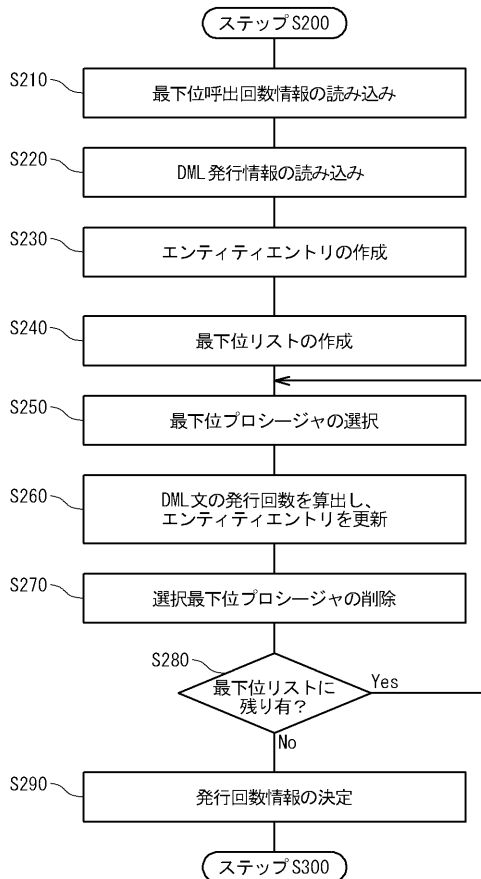
【図 9】



【図 10】



【図 11】



【図 12】

外部呼出回数情報 11

外部処理名	呼出回数
サービス SA	3
サービス SB	10
サービス SC	25

【図 13】

プロシージャ呼出情報 12-1

	サービス SA	サービス SB	サービス SC
プロシージャ P1A	○		
プロシージャ P1B	○	○	
プロシージャ P1C		○	○
プロシージャ P1D			○

【図 14】

DML発行情報 20

	プロシージャ P1A	プロシージャ P1B	プロシージャ P1C	プロシージャ P1D
エンティティ	2	2	2	2

【図 15】

DML負荷情報 30

	DML
エンティティ	6ms

【図 16】

最下位呼出回数情報 15

	呼出回数
プロシージャ P1A	3
プロシージャ P1B	13
プロシージャ P1C	35
プロシージャ P1D	25

【図 17】

発行回数情報 25

	DML文発行回数
エンティティ	152

【図 18】

推定負荷情報 RES

CPU使用率=91.2%
--------------

【図 20】

DML負荷情報 30

	select	update	insert	delete
エンティティ E1	2			
エンティティ E2	3			
エンティティ E3	2	14		
エンティティ E4	3		7	10

【図 19】

DML発行情報 20

	プロシージャ P1A	プロシージャ P1B	プロシージャ P1C	プロシージャ P1D
エンティティ E1	R			R
エンティティ E2		R		
エンティティ E3			RU	
エンティティ E4	D	C		R

【図 21】

発行回数情報 25

	select	update	insert	delete
エンティティ E1	28			
エンティティ E2	13			
エンティティ E3	35	35		
エンティティ E4	25		13	3

【 図 2 2 】

推定負荷情報 RES

	select	update	insert	delete	total
エンティティ E1	56	0	0	0	56
エンティティ E2	39	0	0	0	39
エンティティ E3	70	490	0	0	560
エンティティ E4	75	0	91	30	196
total	240	490	91	30	851

## フロントページの続き

- (56)参考文献 特開平06-028405(JP,A)  
特開昭63-259740(JP,A)  
特開2002-183416(JP,A)  
特開平02-219138(JP,A)  
特開平04-038522(JP,A)  
特開平03-108036(JP,A)  
特開2004-220453(JP,A)  
特開平05-151031(JP,A)  
特開2000-207255(JP,A)  
特開平11-265306(JP,A)  
特開平06-139065(JP,A)  
特開昭61-245273(JP,A)

大串 守、坂東和彦、データベース処理応答時間見積装置、東芝技術公開集、日本、株式会社東芝、2001年 1月15日、VOL.19-2、pp.43~45、(2001-0068)

上坂利文、特集2 最適なインフラ設計のためのキャパシティ・プランニング Part2 キャパシティ・プランニングの実践手法、ITアーキテクト、日本、株式会社アイ・ディ・ジー・ジャパン、2007年10月 2日、Vol.13、pp.110~117

## (58)調査した分野(Int.Cl., DB名)

G06F 11/34  
G06F 11/36