

(19) 日本国特許庁(JP)

(12) 特 許 公 報(B2)

(11) 特許番号

特許第5967672号
(P5967672)

(45) 発行日 平成28年8月10日 (2016. 8. 10)

(24) 登録日 平成28年7月15日 (2016. 7. 15)

(51) Int. Cl. F I
G06F 12/00 (2006.01) G O 6 F 12/00 5 1 3 D
G06F 9/50 (2006.01) G O 6 F 9/46 4 6 2 A

請求項の数 20 (全 43 頁)

(21) 出願番号	特願2014-512275 (P2014-512275)	(73) 特許権者	000005108 株式会社日立製作所 東京都千代田区丸の内一丁目6番6号
(86) (22) 出願日	平成24年4月27日 (2012. 4. 27)	(73) 特許権者	504137912 国立大学法人 東京大学 東京都文京区本郷七丁目3番1号
(86) 国際出願番号	PCT/JP2012/061443	(74) 代理人	110000279 特許業務法人ウィルフォート国際特許事務所
(87) 国際公開番号	W02013/161080	(72) 発明者	徳田 晴介 東京都千代田区丸の内一丁目6番6号 株式会社日立製作所内
(87) 国際公開日	平成25年10月31日 (2013. 10. 31)	(72) 発明者	清水 晃 東京都千代田区丸の内一丁目6番6号 株式会社日立製作所内
審査請求日	平成26年10月22日 (2014. 10. 22)		最終頁に続く

(54) 【発明の名称】 データベース管理システム、計算機、データベース管理方法

(57) 【特許請求の範囲】

【請求項1】

データベースを管理するデータベース管理システムであって、
 前記データベースへのクエリを受け付けるクエリ受付部と、
 前記受け付けたクエリを実行するために必要な1以上のデータベースオペレーションを表す情報を含むクエリ実行プランを生成するクエリ実行プラン生成部と、
 前記生成したクエリ実行プランに基づいて前記受け付けたクエリを実行し、前記受け付けたクエリの実行において、データベースオペレーションを実行するためのタスクを動的に生成し、前記動的に生成されたタスクを実行するクエリ実行部と
 を有し、

前記クエリ実行部は、前記受け付けたクエリの実行において、前記タスクを新たに生成する場合に、生成タスク数決定処理を実行し、

前記生成タスク数決定処理は、新たに生成するタスク当たりに割り当てを要するメモリ資源の量である第1のメモリ資源量と、新たに割り当て可能なメモリ資源の量である第2のメモリ資源量とを基に、生成することができるタスクの数である生成タスク数を算出することであり、

動的に生成されるタスクの数は、前記生成タスク数以下であり、

前記生成タスク数決定処理は、下記(A)及び(B)を含み、

(A) 新たに生成可能なタスクの数である生成可能タスク数を特定すること、

(B) 前記特定された生成可能タスク数と、前記第1のメモリ資源量とを基に、前記特定

された生成可能タスク数分のタスクに割り当てが必要なメモリ資源量である第3のメモリ資源量を算出すること、

前記クエリ実行部は、メモリ資源を予約する処理である予約処理を実行し、

前記予約処理は、クエリの実行に割り当て可能なメモリ資源から、前記第2及び第3のメモリ資源量のうち小さい方のメモリ資源量である予約メモリ資源量分のメモリ資源を予約することであり、

前記生成タスク数決定処理は、更に、下記(C)を含む、

(C)前記第1のメモリ資源量と前記予約メモリ資源量とを基に前記生成タスク数を算出すること、

データベース管理システム。

10

【請求項2】

前記クエリ実行部は、

1以上のクエリを並行して実行する際に、前記1以上のクエリの実行において、別々のタスクにより並行して実行を開始できる実行単位である部分クエリを2以上含み、前記2以上の部分クエリを並行して実行する場合、

前記2以上の部分クエリの各々の実行において、タスクを新たに生成する際に、前記生成タスク数決定処理を行う、

請求項1記載のデータベース管理システム。

【請求項3】

データベースを管理するデータベース管理システムであって、

20

前記データベースへのクエリを受け付けるクエリ受付部と、

前記受け付けたクエリを実行するために必要な1以上のデータベースオペレーションを表す情報を含むクエリ実行プランを生成するクエリ実行プラン生成部と、

前記生成したクエリ実行プランに基づいて前記受け付けたクエリを実行し、前記受け付けたクエリの実行において、データベースオペレーションを実行するためのタスクを動的に生成し、前記動的に生成されたタスクを実行するクエリ実行部とを有し、

前記クエリ実行部は、前記受け付けたクエリの実行において、前記タスクを新たに生成する場合に、生成タスク数決定処理を実行し、

前記生成タスク数決定処理は、新たに生成するタスク当たり割り当てを要するメモリ資源の量である第1のメモリ資源量と、新たに割り当て可能なメモリ資源の量である第2のメモリ資源量とを基に、生成することができるタスクの数である生成タスク数を算出することであり、

30

動的に生成されるタスクの数は、前記生成タスク数以下であり、

前記クエリ実行部は、

1以上のクエリを並行して実行する際に、前記1以上のクエリの実行において、別々のタスクにより並行して実行を開始できる実行単位である部分クエリを2以上含み、前記2以上の部分クエリを並行して実行する場合、

前記2以上の部分クエリの各々の実行において、タスクを新たに生成する際に、前記生成タスク数決定処理を行う、

40

データベース管理システム。

【請求項4】

前記2以上の部分クエリのうちの一つの部分クエリの実行における前記生成タスク数決定処理において、前記第2のメモリ資源量は、第4及び第5のメモリ資源量のうちの小さい方であり、

前記第4のメモリ資源量は、前記2以上の部分クエリにそれぞれ対応した優先度に応じて、全ての部分クエリの実行に割り当て可能なメモリ資源の総量である総メモリ資源量を配分することにより得られる、前記一つの部分クエリの実行に割り当て可能なメモリ資源の量から、前記一つの部分クエリに対応した前記予約メモリ資源量の合計量である既予約メモリ資源量を減じた量であり、

50

前記第5のメモリ資源量は、前記2以上の部分クエリの実行に対応した2以上の前記既予約メモリ資源量の合計を前記総メモリ資源量から減じることにより得られる未予約メモリ資源量である、

請求項2又は3記載のデータベース管理システム。

【請求項5】

前記クエリ実行部は、前記部分クエリ毎の目標実行時間と実行達成率とクエリ実行時間とのうち少なくとも一つを基に、前記部分クエリの前記優先度を動的に変更する、

請求項4記載のデータベース管理システム。

【請求項6】

前記クエリ実行部は、前記部分クエリの前記優先度の変更を受け付ける入力インタフェースで受け付けた情報に従い、前記部分クエリの前記優先度を変更する、

請求項4又は5記載のデータベース管理システム。

【請求項7】

前記クエリ実行部は、

生成されたタスクを実行する場合に、前記予約処理により予約されたメモリ資源から前記第1のメモリ資源量に基づく量のメモリ資源を、前記生成されたタスクに割り当て、

前記生成されたタスクの実行を終了する場合に、前記生成されたタスクに割り当てられたメモリ資源の解放及び予約の解消を行う

請求項1記載のデータベース管理システム。

【請求項8】

前記生成タスク数決定処理は、前記実行されたタスクに対応したデータベースオペレーションの実行結果に基づいて前記タスクを新たに生成する場合に、実行される、

請求項1乃至7のうちのいずれか1項に記載のデータベース管理システム。

【請求項9】

前記クエリ実行部は、前記タスクを新たに生成する場合に、コンテキストを生成して、前記生成したコンテキストに基づいて前記生成タスク数決定処理を実行し、前記生成したコンテキストに基づいて前記生成されたタスクを実行し、

前記コンテキストは、前記新たに生成するタスクにおいて実行を開始するデータベースオペレーションが、前記クエリ実行プランに含まれる情報である1以上のデータベースオペレーションのうちのいずれであることを示す第1の情報と、前記第1の情報が示すデータベースオペレーションに要するデータのアクセス先に関する第2の情報と、前記新たに生成するタスクにより前記1以上のデータベースオペレーションに関する結果を生成するために必要なデータに関する第3の情報とを含む情報である、

請求項1乃至8のうちのいずれか1項に記載のデータベース管理システム。

【請求項10】

前記クエリ実行部は、前記第2のメモリ資源量が増加した場合、前記生成したコンテキストを用いて、前記生成タスク数決定処理を実行する、

請求項9記載のデータベース管理システム。

【請求項11】

前記クエリ実行部は、前記1以上のデータベースオペレーションにおいて、結果生成に至るデータベースオペレーションの総数に対する、前記新たに生成するタスクにおいて実行を開始するデータベースオペレーションから結果生成に至る後続のデータベースオペレーションの数、の比率が、所定値より大きい場合に、前記コンテキストを生成する、

請求項10記載のデータベース管理システム。

【請求項12】

前記クエリ実行部は、前記1以上のデータベースオペレーションにおいて、結果生成に至るデータベースオペレーションの総数に対する、前記新たに生成するタスクにおいて実行を開始するデータベースオペレーションから結果生成に至る後続のデータベースオペレーションの数、の比率が、所定値以下の場合に、前記コンテキスト及び新たなタスクを生成せず、実行しているタスクで前記後続のデータベースオペレーションを実行する、

10

20

30

40

50

請求項 10 記載のデータベース管理システム。

【請求項 13】

前記第 1 のメモリ資源量は、新たに生成するタスクあたりに割り当てを要するメモリ資源の量であって、前記新たに生成するタスクにおいて実行を開始するデータベースオペレーションから後続のデータベースオペレーションの実行に要するメモリ資源量である、請求項 1 乃至 12 のうちのいずれか 1 項に記載のデータベース管理システム。

【請求項 14】

前記クエリ実行部は、前記生成したタスクを実行する場合に、前記実行するタスクに前記第 1 のメモリ資源量に基づく量のメモリ資源を割り当て、前記後続のデータベースオペレーションを実行する、

10

請求項 13 記載のデータベース管理システム。

【請求項 15】

メモリと、
前記メモリに接続された制御デバイスと
を有し、

前記制御デバイスが、
(P) データベースへのクエリを受け付け、
(Q) 前記受け付けたクエリを実行するために必要な 1 以上のデータベースオペレーションを表す情報を含むクエリ実行プランを生成し、

(R) 前記生成したクエリ実行プランに基づいて前記受け付けたクエリを実行し、前記受け付けたクエリの実行において、データベースオペレーションを実行するためのタスクを動的に生成し、前記動的に生成されたタスクを実行し、

20

前記制御デバイスは、前記クエリの実行において、前記タスクを新たに生成する場合に、生成タスク数決定処理を実行し、

前記生成タスク数決定処理は、新たに生成するタスクあたりに割り当てを要するメモリ資源の量である第 1 のメモリ資源量と、新たに割り当て可能なメモリ資源の量である第 2 のメモリ資源量とを基に、生成することができるタスクの数である生成タスク数を算出することであり、

動的に生成されるタスクの数が、前記生成タスク数以下であり、
前記生成タスク数決定処理は、下記 (A) 及び (B) を含み、

30

(A) 新たに生成可能なタスクの数である生成可能タスク数を特定すること、
(B) 前記特定された生成可能タスク数と、前記第 1 のメモリ資源量とを基に、前記特定された生成可能タスク数分のタスクに割り当てが必要なメモリ資源量である第 3 のメモリ資源量を算出すること、

前記制御デバイスは、メモリ資源を予約する処理である予約処理を実行し、
前記予約処理は、クエリの実行に割り当て可能なメモリ資源から、前記第 2 及び第 3 のメモリ資源量のうち小さい方のメモリ資源量である予約メモリ資源量分のメモリ資源を予約することであり、

前記生成タスク数決定処理は、更に、下記 (C) を含む、
(C) 前記第 1 のメモリ資源量と前記予約メモリ資源量とを基に前記生成タスク数を算出すること、

40

計算機。

【請求項 16】

メモリと、
前記メモリに接続された制御デバイスと
を有し、

前記制御デバイスが、
(P) データベースへのクエリを受け付け、
(Q) 前記受け付けたクエリを実行するために必要な 1 以上のデータベースオペレーションを表す情報を含むクエリ実行プランを生成し、

50

(R) 前記生成したクエリ実行プランに基づいて前記受け付けたクエリを実行し、前記受け付けたクエリの実行において、データベースオペレーションを実行するためのタスクを動的に生成し、前記動的に生成されたタスクを実行し、

前記制御デバイスは、前記クエリの実行において、前記タスクを新たに生成する場合に、生成タスク数決定処理を実行し、

前記生成タスク数決定処理は、新たに生成するタスク当たり割り当てを要するメモリ資源の量である第1のメモリ資源量と、新たに割り当て可能なメモリ資源の量である第2のメモリ資源量とを基に、生成することができるタスクの数である生成タスク数を算出することであり、

動的に生成されるタスクの数が、前記生成タスク数以下であり、

前記制御デバイスは、

1以上のクエリを並行して実行する際に、前記1以上のクエリの実行において、別々のタスクにより並行して実行を開始できる実行単位である部分クエリを2以上含み、前記2以上の部分クエリを並行して実行する場合、

前記2以上の部分クエリの各々の実行において、タスクを新たに生成する際に、前記生成タスク数決定処理を行う、
計算機。

【請求項17】

前記2以上の部分クエリのうちの一部分クエリの実行における前記生成タスク数決定処理において、前記第2のメモリ資源量は、第4及び第5のメモリ資源量のうちの小さい方であり、

前記第4のメモリ資源量は、前記2以上の部分クエリにそれぞれ対応した優先度に応じて、全ての部分クエリの実行に割り当て可能なメモリ資源の総量である総メモリ資源量を配分することにより得られる、前記一部分クエリの実行に割り当て可能なメモリ資源の量から、前記一部分クエリに対応した前記予約メモリ資源量の合計量である既予約メモリ資源量を減じた量であり、

前記第5のメモリ資源量は、前記2以上の部分クエリの実行に対応した2以上の前記既予約メモリ資源量の合計を前記総メモリ資源量から減じることにより得られる未予約メモリ資源量である、

請求項16記載の計算機。

【請求項18】

前記制御デバイスは、前記タスクを新たに生成する場合に、コンテキストを生成して、前記生成したコンテキストに基づいて、前記生成タスク数決定処理を実行し、前記生成したタスクを実行し、

前記コンテキストは、前記新たに生成するタスクにおいて実行を開始するデータベースオペレーションが、前記クエリ実行プランに含まれる情報である1以上のデータベースオペレーションのうちいずれであるかを示す第1の情報と、前記第1の情報が示すデータベースオペレーションに要するデータのアクセス先に関する第2の情報と、新たに生成するタスクにより前記1以上のデータベースオペレーションに関する結果を生成するために必要なデータに関する第3の情報とを含む情報である、

請求項15乃至17のうちのいずれか1項に記載の計算機。

【請求項19】

データベースを管理するデータベース管理方法であって、

(a) 前記データベースへのクエリを受け付け、

(b) 前記受け付けたクエリを実行するために必要な1以上のデータベースオペレーションを表す情報を含むクエリ実行プランを生成し、

(c) 前記生成したクエリ実行プランに基づいて前記受け付けたクエリを実行し、前記受け付けたクエリの実行において、データベースオペレーションを実行するためのタスクを動的に生成し、前記動的に生成されたタスクを実行し、

前記(c)では、前記クエリの実行において、前記タスクを新たに生成する場合に、生

10

20

30

40

50

成タスク数決定処理を実行し、

前記生成タスク数決定処理は、新たに生成するタスク当たり割り当てを要するメモリ資源量である第1のメモリ資源量と、新たに割り当て可能なメモリ資源量である第2のメモリ資源量とを基に、生成することができるタスクの数である生成タスク数を算出することであり、

動的に生成されるタスクの数は、前記生成タスク数以下であり、

前記生成タスク数決定処理は、下記(A)及び(B)を含み、

(A) 新たに生成可能なタスクの数である生成可能タスク数を特定すること、

(B) 前記特定された生成可能タスク数と、前記第1のメモリ資源量とを基に、前記特定された生成可能タスク数分のタスクに割り当てが必要なメモリ資源量である第3のメモリ資源量を算出すること、

10

前記(c)では、メモリ資源を予約する処理である予約処理を実行し、

前記予約処理は、クエリの実行に割り当て可能なメモリ資源から、前記第2及び第3のメモリ資源量のうち小さい方のメモリ資源量である予約メモリ資源量分のメモリ資源を予約することであり、

前記生成タスク数決定処理は、更に、下記(C)を含む、

(C) 前記第1のメモリ資源量と前記予約メモリ資源量とを基に前記生成タスク数を算出すること、

データベース管理方法。

【請求項20】

20

データベースを管理するデータベース管理方法であって、

(a) 前記データベースへのクエリを受け付け、

(b) 前記受け付けたクエリを実行するために必要な1以上のデータベースオペレーションを表す情報を含むクエリ実行プランを生成し、

(c) 前記生成したクエリ実行プランに基づいて前記受け付けたクエリを実行し、前記受け付けたクエリの実行において、データベースオペレーションを実行するためのタスクを動的に生成し、前記動的に生成されたタスクを実行し、

前記(c)では、前記クエリの実行において、前記タスクを新たに生成する場合に、生成タスク数決定処理を実行し、

前記生成タスク数決定処理は、新たに生成するタスク当たり割り当てを要するメモリ資源量である第1のメモリ資源量と、新たに割り当て可能なメモリ資源量である第2のメモリ資源量とを基に、生成することができるタスクの数である生成タスク数を算出することであり、

30

動的に生成されるタスクの数は、前記生成タスク数以下であり、

1以上のクエリを並行して実行する際に、前記1以上のクエリの実行において、別々のタスクにより並行して実行を開始できる実行単位である部分クエリを2以上含み、前記2以上の部分クエリを並行して実行する場合、前記2以上の部分クエリの各々の実行において、タスクを新たに生成する際に、前記生成タスク数決定処理を行う、

データベース管理方法。

40

【発明の詳細な説明】

【技術分野】

【0001】

本発明は、データベース管理技術に関する。

【背景技術】

【0002】

企業活動において、大量に生じる業務データの活用は不可欠になっている。そのため、大量の業務データを蓄積したデータベース(以下、「DB」)を解析処理するシステムが既に考案されている。

【0003】

50

この解析処理において、データベース管理システム（以下、「DBMS」）は、クエリを受け付け、DBを格納する記憶デバイスにデータ読出し要求を発行する。

【0004】

一つのクエリの実行におけるデータ読出しの待ち時間の短縮化を図る技術として、特許文献1に開示の技術が知られている。特許文献1によれば、DBMSは、クエリ実行に要するデータ読出しの都度にタスクを動的に生成し並行実行することでデータ読出し要求を多重化する。DBMSは、動的に生成されたタスクに、そのタスクにより実行されるデータベースオペレーション（以下、「DBオペレーション」）に要するメモリ資源を割り当てる。特許文献1によれば、DBMSは、現存するタスクの数と所定数とを比較し、現存するタスクの数が所定数に達したらタスクの生成を待つ。

10

【先行技術文献】

【特許文献】

【0005】

【特許文献1】特開2007-34414号公報

【発明の概要】

【発明が解決しようとする課題】

【0006】

クエリを実行する際に消費することができる最大のメモリ資源の量（割り当て可能なメモリ資源量）に制約を課すシステム運用が考えられる。一つのクエリの実行に使用されるメモリ資源量は、動的に生成されるタスクで実行するDBオペレーションや生成するタスクの数に依存し、且つ、クエリの実行の進行に従い変化する。更に、複数のクエリを同時に実行する場合に使用されるメモリ資源量は、各クエリの実行の進行における時間的な重なり具合に応じて変化する。そのため、割り当て可能なメモリ資源量に制約を課す場合、各クエリを並行処理するタスクの上限数を、該タスクにより実行されるDBオペレーションに応じて適切に決定することが望ましい。

20

【0007】

例えば、特許文献1の技術が適用された環境において、DBMSは、動的に生成するタスクに、該タスクにより実行されるDBオペレーションに必要なメモリ資源を順次割り当てると仮定する。その環境において、DBMSが1以上のクエリを同時に実行する際に、現存するタスクの数と比較される前述の所定数として、不適切な値が設定されたとする。このとき、以下の(1)及び/又は(2)の問題が生じ得る。

30

(1) DBMSは、クエリの実行結果を生成する前に多くのタスクを生成し、割り当て可能なメモリ資源の量を超える大量のメモリ資源を消費してしまう。これにより、メモリ資源が枯渇してスラッシングが発生し、システム全体のスローダウンが引き起こされる。

(2) 割り当て可能なメモリ資源の量に余裕があるにも関わらず、DBMSにより並行して実行されるタスクの数が少なくなってしまう。これにより、データ読出し要求の多重度が不足し、その結果、性能を十分に引き出すことができない。

そこで、本発明の目的は、割り当て可能なメモリ資源量に制約のある環境において、クエリ実行時に動的に生成されるタスクの数を適切な数とし、当該制約の範囲内でクエリ実行時間を短縮化することにある。

40

【課題を解決するための手段】

【0008】

DBMSが、クエリ受付部と、クエリ実行プラン生成部と、クエリ実行部とを有する。例えば、DBMSは、コンピュータプログラムであって、計算機で実行されることにより、クエリ受付部と、クエリ実行プラン生成部と、クエリ実行部とを計算機に構築する。

【0009】

クエリ受付部は、クエリを受け付ける。クエリ実行プラン生成部は、クエリを実行するために必要な1以上のDBオペレーションを表す情報を含むクエリ実行プランを生成する。クエリ実行部は、生成したクエリ実行プランに基づいて、受け付けたクエリを実行する。

50

【0010】

クエリ実行部は、クエリの実行において、DBオペレーションを実行するためのタスクを動的に生成し、動的に生成されたタスクを実行する。具体的には、例えば、クエリ実行部は、クエリの実行において、(a) DBオペレーションを実行するためのタスクを生成すること、(b) 生成されたタスクを実行することで、そのタスクに対応したDBオペレーションに必要なデータを読み出すためにDBヘデータ読出し要求を発行すること、(c) 上記(b)で実行されたタスクに対応したN番目のDBオペレーションの実行結果に基づき(N+1)番目のDBオペレーションを実行する場合には、その実行結果に基づくタスクを新たに生成すること(Nは1以上の整数)、及び、(d) その新たに生成したタスクについて上記(b)及び上記(c)を行うこと、を行い、上記(b)及び(d)において、2以上の実行可能なタスクが存在する場合には、それら2以上のタスクのうち少なくとも二つのタスクを並行して実行するようになっている。このクエリ実行部の動作は、前述した特許文献1に開示の技術に従う動作で良い。

10

【0011】

クエリ実行部は、クエリの実行において、タスクを新たに生成する場合に(例えば上記(a)又は(c)の場合に)、同時生成タスク数決定処理を行う。同時生成タスク数決定処理は、新たに生成可能なタスクの数である生成可能タスク数と、新たに生成するタスク当たり割り当てを要するメモリ資源の量である第1のメモリ資源量と、新たに割り当て可能なメモリ資源の量である第2のメモリ資源量とを基に、同時に生成可能なタスクの数である同時生成タスク数を算出することである。動的に同時に生成されるタスクの数は、算出した同時生成タスク数以下である。

20

【発明の効果】

【0012】

割り当て可能なメモリ資源量が限られる環境において、クエリ実行時に動的に生成されるタスクの数を適切な数とすることができる。具体的には、割り当て可能なメモリ資源の量に制約のある環境において、当該制約の範囲内においてデータ読出し要求を最も高い多重度を以って発行することが期待でき、故に、クエリ実行時間の短縮化が期待できる。

【図面の簡単な説明】

【0013】

【図1】図1は、実施例1に係る索引A及び表Aを示す。

30

【図2】図2は、実施例1に係る索引B及び表Bを示す。

【図3】図3は、実施例1に係るクエリ1を示す。

【図4】図4は、実施例1に係るクエリ2を示す。

【図5】図5は、実施例1に係るクエリ1の実行プランを示す。

【図6】図6は、実施例1に係るクエリ2の実行プランを示す。

【図7】図7は、メモリ資源が枯渇することを説明するための模式図の一例である。

【図8】図8は、実施例1でクエリ1の実行においてメモリ資源の枯渇を回避することを説明するための模式図の一例である。

【図9】図9は、実施例1でクエリ1及びクエリ2の同時実行においてメモリ資源の枯渇を回避することを説明するための模式図の一例である。

40

【図10】図10は、実施例1に係る計算機システムの構成を示す。

【図11】図11は、実施例1に係るクエリ実行管理表の構成を示す。

【図12】図12は、実施例1に係るクエリ実行の全体の流れを示す。

【図13】図13は、実施例1に係るタスク実行処理の流れを示す。

【図14】図14は、実施例1に係るタスク生成保留処理の流れを示す。

【図15】図15は、実施例1に係る同時生成タスク数決定処理の流れを示す。

【図16】図16は、実施例1に係るメモリ資源割り当て処理の流れを示す。

【図17】図17は、実施例1に係るメモリ資源解放処理の流れを示す。

【図18】図18は、実施例1に係るメモリ資源予約処理の流れを示す。

【図19】図19は、実施例1に係るメモリ資源増加処理の流れを示す。

50

【図20】図20は、実施例1に係るメモリ資源減少処理の流れを示す。

【図21】図21は、実施例1に係るサーバメモリ資源増加処理の流れを示す。

【図22】図22は、実施例1に係るサーバメモリ資源減少処理の流れを示す。

【図23】図23は、実施例1に係る優先度手動変更処理の流れを示す。

【図24】図24は、実施例1に係る優先度自動変更処理の流れを示す。

【図25】図25は、実施例1に係るタスク追加生成処理の流れを示す。

【図26】図26は、実施例2でクエリ1の実行においてメモリ資源の枯渇を回避することを説明するための模式図の一例である。

【図27】図27は、実施例2でクエリ1及びクエリ2の同時実行においてメモリ資源の枯渇を回避することを説明するための模式図の一例である。

10

【図28】図28は、実施例2に係るクエリ実行部の構成を示す。

【図29】図29は、実施例2に係るクエリ実行の全体の流れを示す。

【図30】図30は、実施例2に係るタスク実行処理の流れを示す。

【図31】図31は、実施例2に係るDBオペレーション処理の流れを示す。

【図32】図32は、実施例2に係るタスク追加生成処理の流れを示す。

【図33】図33は、実施例3に係る計算機システムの構成を示す。

【発明を実施するための形態】

【0014】

以下、図面を参照しながら、幾つかの実施例を説明する。なお、以下の説明により本発明が限定されるものではない。また、以下の説明では、データベースを「DB」、データベース管理システムを「DBMS」、DBMSを実行するサーバを「DBサーバ」と言う。DBMSに対するクエリの発行元は、DBMSの外部のコンピュータプログラム（例えばアプリケーションプログラム）で良い。外部のコンピュータプログラムは、DBサーバ内で実行されるプログラムでも良いし、DBサーバに接続された装置（例えばクライアント計算機）で実行されるプログラムでも良い。

20

【実施例1】

【0015】

まず、本実施例の概要を説明する。

【0016】

DBサーバは、DBMSを実行する。DBMSは、クエリを受け付け、受け付けたクエリを実行する。そして、その実行により生成された結果をクエリの発行元に返す。クエリの結果を生成するまでには、DBMSは、1以上のDBオペレーションを実行する。DBMSは、それらDBオペレーションのうち少なくとも一つのDBオペレーションの実行において、DBを格納した記憶デバイスに対して読み出し要求を発行する場合がある。

30

【0017】

例えば、DBMSは、図1及び図2に示す索引A、表A、索引B及び表Bを含むDBを記憶デバイス（例えば、DBサーバと通信できるように接続された外部ストレージ装置）に格納しているとする。表は、1以上のレコードの集合であり、レコードは、1以上のカラムから構成される。索引は、表の中の1以上のカラムを対象に作成されるデータ構造であり、当該索引が対象とするカラムを含む選択条件による表へのアクセスを高速化する。例えば、索引は、対象とするカラムの値毎にその値を含む表の中のレコードを特定する情報（RowID）を保持するデータ構造であり、B木構造等が用いられる。

40

【0018】

例えば、DBMSは、索引Aを用いて、表AのカラムA_Typeの値が「AAA」であるレコードに対応したRowIDの集合であるRowID List「a1」から、表Aの二つのレコード（1レコード目及び3レコード目）を特定することができる。また、表AのカラムAC2における値と表BのカラムBC1における値とが互いに関連しているとする。このとき、DBMSは、カラムA_Typeの特定の値における索引AのRowID Listを用いて、表AからA_Typeの該当値を含むレコードを特定する。そして、特定したレコードに含まれるカラムAC1及びAC2等のそれぞれの値を得る。更

50

に、DBMSは、索引BのRowID Listを用いて、先に得たAC2の値と関連する表BのカラムBC1の値を含むレコードを特定する。こうしてDBMSは、表Aと表Bそれぞれのカラムの値を関連付けることで、先に特定した表Bのレコードに含まれるカラムBC2等の値を得ることができる。

【0019】

例えば、DBMSが受け付けたクエリは、図3に示すクエリ1と、図4に示すクエリ2であるとする。クエリ1は、表AのカラムA_Typeの値が「AAA」であり、且つ、表AのカラムAC2の値と表BのカラムBC1の値とが一致する表Aと表Bのレコードにおいて、表AのカラムAC1の値と表BのカラムBC2の値を抽出するクエリである。同様に、クエリ2は、表AのカラムA_Typeの値が「BBB」であり、且つ、表AのカラムAC2の値と表BのカラムBC1の値とが一致する表Aと表Bのレコードにおいて、表AのカラムAC1の値と表BのカラムBC2の値を抽出するクエリである。

10

【0020】

DBMSは、上記したクエリ1及びクエリ2を実行するために、例えば、図5及び図6に示すようなクエリ実行プランをそれぞれ生成する。クエリ実行プランは、例えば、データ読出しを伴う一つ以上のDBオペレーションを表す情報を含む。クエリ実行プランにおけるDBオペレーションの実行順序には木構造の順序関係が存在する。DBMSは、クエリ1又は2のクエリ実行プランに基づき、索引Aを用いて、表AのカラムA_Typeの指定値を含んだレコードにおけるカラムAC1とAC2の値を抽出する。更に、索引Bを用いて、抽出したAC1の値と一致する表BのカラムBC1の値を含んだレコードにおけるカラムBC2の値を抽出する。DBMSは、クエリ実行の結果として、抽出された値、すなわち、AC1の値とBC2の値を生成する。具体的には、DBMSは、下記の処理を行う。

20

(S1) 索引Aを用いて、カラムA_Typeの指定値を含む表Aのレコードに対応したRowID Listを検索する、

(S2) 上記(S1)で見つかったRowID Listを用いて表Aの該当するレコードを含むデータをフェッチし、該当するレコードのカラムAC1及びAC2の値を抽出する、

(S3) 索引Bを用いて、上記(S2)で抽出したカラムAC2の値と一致するカラムBC1の値を含んだ表BのレコードのRowID Listを検索する、

30

(S4) 上記(S3)で見つかったRowID Listを用いて表Bの該当するレコードを含むデータをフェッチし、該当するレコードのカラムBC2の値を抽出する、

(S5) クエリ実行の結果として、抽出したカラムAC1及びBC2の値を生成しクエリ発行元に返す。

【0021】

上述の通り、DBMSは、クエリ実行プランに従ってクエリを実行する。DBMSが、図3のクエリ1を図5のクエリ実行プランに従って実行する際に、その実行に消費することができる最大のメモリ資源の量(割り当て可能なメモリ資源量)を考慮することなくタスクを動的に生成していくと、図7に例示する問題(メモリ資源が枯渇しスラッシングが発生すること)が生じ得る。以下、その問題を説明する。なお、図7の表記は、次のルールに従う。

40

(*) 横軸は、時刻を表す。

(*) 図上部の横に長い五角形は、一つのタスクによるDBオペレーションを意味する。五角形の左端は、タスクが生成され、該タスクによるDBオペレーションが開始される時刻を表す。五角形の右端は、該タスクによるDBオペレーションが終了し、該タスクが終了される時刻を表す。

(*) 図上部の五角形の内部の値は、タスクに対応したDBオペレーションに伴いフェッチされるデータと、結果を生成するために必要なフェッチ済みのデータとを表す。

(*) 図下部の縦軸は、クエリを実行する際に消費されるメモリ資源の量(割り当てられたメモリ資源の量)を表す。

50

(*) 割り当て可能なメモリ資源量の上限値(以下、割り当て上限値)は「6」であるとする。

(*) 一つのタスクに対応するDBオペレーションに必要なメモリ資源量は「1」であるとする。なお、タスク自体の生成に要するメモリ資源は、クエリの実行に消費されるメモリ資源とは別に管理されているものとする。

【0022】

特許文献1の技術によれば、タスクにより実行されるDBオペレーションの結果に基づき1以上のタスクを動的に生成することができる。図7の例によれば、DBMSは以下のようにクエリ1を実行する。

(t0) 索引Aにアクセスするためのタスク11Aを生成する。タスク11Aでは、表AのカラムA_Typeの値が「AAA」であるレコードのRowID Listの検索を行う。DBMSは、タスク11Aの実行に必要なメモリ資源を割り当て、タスク11Aを実行する。

(t1) タスク11Aの実行によりRowID List「a1」を得る。DBMSは、その実行結果に基づき、表Aのデータをフェッチするためのタスク11B及び11Cを生成する。タスク11Bでは、表Aの1レコード目を含むデータをフェッチし、タスク11Cでは、表Aの3レコード目を含むデータをフェッチする。DBMSは、タスク11B及び11Cの実行に必要なメモリ資源をそれぞれ割り当て、タスク11B及び11Cを実行する。その後、タスク11Aに割り当てていたメモリ資源を解放しタスク11Aを終了する。

(t2) タスク11Bの実行により、フェッチされた表Aの1レコード目を含むデータからAC1の値「A1」及びAC2の値「001」を抽出する。DBMSは、その実行結果に基づき、索引Bにアクセスするためのタスク11Xを生成する。タスク11Xでは、抽出したAC2の値と一致するBC1の値を持つ表BのレコードのRowID Listの検索を行う。DBMSは、タスク11Xの実行に必要なメモリ資源を割り当て、タスク11Xを実行する。なお、DBMSは、タスク11Xに割り当てたメモリ資源に、クエリの結果を生成するためのデータであるAC1の値「A1」、及び、タスク11Xに対応するDBオペレーション(索引BへのRowID Listの検索)を行うために必要なデータであるAC2の値「001」を、それぞれ設定する。その後、DBMSは、タスク11Bに割り当てていたメモリ資源を解放しタスク11Bを終了する。同様に、DBMSは、タスク11Cの実行により、フェッチされた表Aの3レコード目を含むデータからAC1の値「A3」及びAC2「003」の値を抽出する。DBMSは、その実行結果に基づき、索引Bにアクセスするためのタスク11Yを生成する。タスク11Yでは、抽出したAC2の値と一致するBC1の値を持つ表BのレコードのRowID Listの検索を行う。DBMSは、タスク11Yの実行に必要なメモリ資源を割り当て、タスク11Yを実行する。なお、DBMSは、タスク11Yに割り当てたメモリ資源に、クエリの結果を生成するためのデータであるAC1の値「A3」、及び、タスク11Yに対応するDBオペレーション(索引BへのRowID Listの検索)を行うために必要なデータであるAC2の値「003」を、それぞれ設定する。その後、DBMSは、タスク11Cに割り当てていたメモリ資源を解放しタスク11Cを終了する。

t2の時点では、クエリの実行に際して消費されるメモリ資源の割り当て量は、割り当て上限値「6」を超えない。

【0023】

しかし、時間の経過に伴い、クエリの実行に割り当て中のメモリ資源の量は変化(増加又は減少)する。割り当て上限値を考慮せずにDBMSがタスクを動的に生成していくと、図7に例示する通り、消費されるメモリ資源の量が、割り当て上限値「6」を上回ってしまう。その結果、メモリ資源が枯渇することがある。図7によれば、DBMSは、t3で以下の処理を行う。

(t3) タスク11Xの実行によりRowID List「b1」を得る。DBMSは、その実行結果に基づき、表Bの三つのレコードを含むデータそれぞれをフェッチするため

10

20

30

40

50

のタスクを三つ生成し、その実行に必要なメモリ資源をそれぞれ割り当てようとする。また、同じく t 3 で、DBMS は、タスク 1 1 Y の実行により Row ID List「b 3」を得る。DBMS は、その実行結果に基づき、表 B の五つのレコードを含むデータそれぞれをフェッチするためのタスクを五つ生成し、その実行に必要なメモリ資源をそれぞれ割り当てようとする。

すなわち、DBMS は、t 3 で、8 個のタスクを生成しそれぞれにメモリ資源を割り当てようとする。しなしながら、割り当て上限値が「6」であるため、タスクに割り当てるメモリ資源が枯渇し、スラッシングが発生してしまう。その結果、システム全体のスローダウンを引き起こしてしまう。ここで、割り当て上限値は、時間の経過に伴い変化しうる。その原因として、例えば、DBMS 以外のコンピュータプログラムの実行状況や、仮想化プログラムが作成し実行する仮想マシン上に DBMS を構築した際における仮想マシンの総メモリ資源量の変化があげられる。

【0024】

そこで、本実施例では、DBMS は、タスクを新たに生成する都度に、同時生成タスク数決定処理を行う。同時生成タスク数決定処理は、新たに生成可能なタスクの数である生成可能タスク数と、新たに生成するタスク当たり割り当てるメモリ資源量である第 1 のメモリ資源量と、新たに割り当て可能なメモリ資源量である第 2 のメモリ資源量とを基に、同時に生成可能なタスクの数である同時生成タスク数を算出することである。本実施例では、第 1 のメモリ資源量は、新たに生成するタスクに対応した DB オペレーションに要するメモリ資源量 (DB オペレーションメモリ資源量) に基づくメモリ資源量である。例えば、第 1 のメモリ資源量は、DB オペレーションメモリ資源量と同じ量、DB オペレーションメモリ資源量よりも多くのメモリ資源量、或いは、他のタスクとメモリ資源を共通化可能等の理由があれば、DB オペレーションメモリ資源量よりも少ないメモリ資源量である。同時に生成されるタスクの数は、同時生成タスク数と同数でなくても良く、同時生成タスク数より少ない数でも良い。

【0025】

図 8 は、DBMS が図 3 のクエリ 1 を図 5 のクエリ実行プランに従って実行する際に、メモリ資源の枯渇を回避できることを示す模式図の例を示す。表記のルールは図 7 と同様である。図 8 によれば、DBMS は、以下の通りクエリ 1 を実行する。

(t 0) 索引 A にアクセスするためのタスクを生成する際に、同時生成タスク数決定処理を行う。例えば、DBMS は、生成可能タスク数「1」と、第 1 のメモリ資源量「1」と、第 2 のメモリ資源量「6」(割り当て上限値「6」)とに基づき、同時生成タスク数を「1」と算出する。DBMS は、算出した同時生成タスク数「1」と同数のタスクを生成し、対応する DB オペレーションに要するメモリ資源を割り当てて実行する。

(t 1) t 0 で実行したタスクの結果に基づいて、表 A の二つのレコードを含むデータそれぞれをフェッチするためのタスクを二つ生成する際に、同時生成タスク数決定処理を行う。例えば、DBMS は、生成可能タスク数「2」と、第 1 のメモリ資源量「1」と、第 2 のメモリ資源量「5」(割り当てて上限値「6」から割り当て済みのメモリ資源量「1」を減じた値)とに基づき、同時生成タスク数を「2」と算出する。DBMS は、算出した同時生成タスク数「2」と同数のタスク 1 1 B' 及び 1 1 C' を生成し、対応する DB オペレーションに要するメモリ資源を割り当てて実行する。

(t 2) タスク 1 1 B' の実行結果に基づいて、索引 B にアクセスするためのタスクを一つ生成する際に、同時生成タスク数決定処理を行う。例えば、DBMS は、生成可能タスク数「1」と、第 1 のメモリ資源量「1」と、第 2 のメモリ資源量「4」(割り当てて上限値「6」から割り当て済みのメモリ資源量「2」を減じた値)とに基づき、同時生成タスク数を「1」と算出する。DBMS は、算出した同時生成タスク数「1」と同数のタスク 1 1 X' を生成し、対応する DB オペレーションに要するメモリ資源を割り当てて実行する。タスク 1 1 C' についても同様に、その実行結果に基づいてタスクを一つ生成する際に、同時生成タスク数決定処理を行い、同時生成タスク数を「1」と算出する。DBMS は、算出した同時生成タスク数「1」と同数のタスク 1 1 Y' を生成し、対応する DB

10

20

30

40

50

オペレーションに要するメモリ資源を割り当てて実行する。

t 2 までは、クエリ実行に消費されるメモリ資源の量が割り当て上限値「6」を超えない。そのため、実行するタスクの振る舞いと、クエリ実行に消費されるメモリ資源の量の変化とは、図7と同じである。

t 3 において、DBMS がタスクを新たに生成する際に、図7と違って、同時に生成し実行されるタスクの数は以下述べるように8個にならず、メモリ資源の枯渇を回避することができる。

(t 3) タスク 1 1 X ' の実行結果に基づいて、表 B の三つのレコードを含むデータそれぞれをフェッチするためのタスクを三つ生成する際に、同時生成タスク数決定処理を行う。例えば、DBMS は、生成可能タスク数「3」と、第1のメモリ資源量「1」と、第2のメモリ資源量「4」(割り当てて上限値「6」から割り当て済みのメモリ資源量「2」を減じた値)とに基づき、同時生成タスク数を「3」と算出する。DBMS は、算出した同時生成タスク数のタスク三つを生成し、対応するDBオペレーションに要するメモリ資源を割り当てて実行する。同様に、タスク 1 1 Y ' の実行結果に基づいて、表 B の五つのレコードを含むデータそれぞれをフェッチするためのタスクを五つ生成する際に、同時生成タスク数決定処理を行う。例えば、DBMS は、生成可能タスク数「5」と、第1のメモリ資源量「1」と、第2のメモリ資源量「1」(割り当て上限値「6」から割り当て済みのメモリ資源量「5」を減じた値)とに基づき、同時生成タスク数を「1」と算出する。DBMS は、算出した同時生成タスク数「1」と同数のタスクを生成し、対応するDBオペレーションに要するメモリ資源を割り当てて実行する。このとき、タスク 1 1 Y ' について、その実行結果に基づいて新たに生成可能なタスクは残り四つあるので、DBMS は、新たなタスクの生成が可能になるまでタスク 1 1 Y ' によるタスクの生成を待たせる。一方、タスク 1 1 X ' について、その実行結果に基づいて生成可能であった三つのタスクを全て生成し実行を開始したので、DBMS は、タスク 1 1 X ' に割り当てていたメモリ資源を解放しタスク 1 1 X ' を終了する(t 3 の直後)。タスク 1 1 X ' の終了により、新たに割り当て可能なメモリ資源量である第2のメモリ資源量が「1」となったため、DBMS は、タスクの生成を待っていたタスク 1 1 Y ' について同時生成タスク数決定処理を行う。該処理により同時生成タスク数を「1」と算出し、算出した数と同数のタスクを生成しメモリ資源を割り当てて実行する。タスク 1 1 Y ' の実行結果に基づいて新たに生成可能なタスクは残り三つあるので、DBMS は、新たなタスクの生成が可能になるまでタスク 1 1 Y ' によるタスクの生成を待たせる。

(t 4) t 3 で実行した四つのタスクについて、その実行を終え、クエリの結果を生成する。DBMS は、実行を終えた四つのタスクそれぞれについて、割り当てていたメモリ資源を解放しタスクを終了する(t 4 の直後)。これにより、第2のメモリ資源量が「4」となったため、DBMS は、タスクの生成を待っていたタスク 1 1 Y ' について同時生成タスク数決定処理を行う。該処理により同時生成タスク数を「3」と算出し、算出した数と同数のタスクを生成しメモリ資源を割り当てて実行する。

(t 5) t 4 までに実行した全てのタスクの実行を終え、クエリの結果を生成する。

【 0 0 2 6 】

このように、実施例1では、DBMS は、新たにタスクを生成する都度に、同時生成タスク数決定処理により同時生成タスク数を決定する。そして、タスクに対応したDBオペレーションの実行結果に基づき、動的に生成するタスクの総数を同時生成タスク数以下とする。これにより、クエリ実行に消費されるメモリ資源の量は、割り当て上限値を超えない。従って、タスクに割り当てるメモリ資源が枯渇してしまうことを回避することができる。同時に生成されるタスクの数を同時生成タスク数と同じにすれば、割り当て上限値の範囲内でデータ読み出し要求を最も高い多重度を以って発行することが期待できる。故に、クエリ実行時間の短縮化が期待できる。なお、「同時に生成されるタスク」とは、或るDBオペレーションの結果に基づき、ほぼ同じ時間帯に生成されるタスクであることを意味する。

【 0 0 2 7 】

10

20

30

40

50

また、実施例 1 では、DBMS が複数のクエリを受け付け、受け付けた複数のクエリを並行して実行する場合にも、タスクに割り当てるためのメモリ資源が枯渇することを回避することができる。DBMS が図 3 のクエリ 1 と図 4 のクエリ 2 を同時に受け付け、その二つのクエリを図 5 及び図 6 のクエリ実行プランに従って並行に実行する際の模式図の例を図 9 に示す。表記のルールは図 7 と同様である。実施例 1 では、DBMS は、実行するクエリ毎に優先度を設ける。図 9 に示すように、それら優先度を基に、DBMS は、割り当て上限値「6」を、実行するクエリ毎に配分する。例えば、DBMS は、優先度が高いほど、割り当て可能なメモリ資源量のうちのより多くの量を割り振る。実施例 1 では、優先度を表す数値が高い方が、優先度が高い。例えば、同時に受け付けたクエリ 1 及びクエリ 2 について、クエリ 1 の優先度がクエリ 2 の優先度よりも 2 倍高い場合、DBMS は、割り当て上限値「6」のうちの「4」を、クエリ 1 の実行について割り当て可能なメモリ資源量の上限値とし、割り当て上限値「6」のうちの「2」を、クエリ 2 の実行について割り当て可能なメモリ資源量の上限値とする。DBMS は、各クエリの実行において、タスクを新たに生成する都度に同時生成タスク数決定処理を行う。すなわち、DBMS は、クエリ毎に、対応する割り当て可能なメモリ資源量の上限値に基づいて、同時生成タスク数を算出する。そして、各クエリの実行において、新たに生成するタスクの数を、算出した同時生成タスク数以下とする。

上記の通り、複数のクエリを並行して実行する際に消費されるメモリ資源の総量を、割り当て上限値以下とし、タスクに割り当てるためのメモリ資源の枯渇を回避することができる。そして、同時に生成されるタスクの数を同時生成タスク数と同じにすることで、割り当て上限値の範囲内において、各クエリのデータ読み出し要求を、クエリ毎の優先度に応じた最大の多重度で発行することが期待できる。故に、クエリ毎の実行時間をクエリ毎の優先度に応じて短縮化することが期待できる。

【0028】

なお、各クエリの実行に割り当て可能なメモリ資源量の上限値は、例えば、下記(1)又は(2)、

(1) 同時に実行するクエリの総数が変化する、

(2) 少なくとも一つのクエリの優先度が変化する、

を契機に変化して良い。

また、図 7 ~ 図 9 は、概要のイメージを表す模式的な図であり、DBMS は、複数のタスクを同じ時点で開始しなくても良い。

【0029】

以下、実施例 1 を詳細に説明する。

【0030】

図 10 は、実施例 1 に係る計算機システムの構成を示す。

【0031】

DBサーバ 401 が通信ネットワーク 403 を介して外部ストレージ装置 402 に接続されている。通信ネットワーク 403 を介した通信のprotocolsとしては、例えば、FC (Fibre Channel)、SCSI (Small Computer System Interface)、又は、TCP/IP (Transmission Control Protocol / Internet Protocol) が採用されて良い。

【0032】

DBサーバ 401 は、計算機、例えば、パーソナルコンピュータ、ワークステーション又はメインフレーム、もしくは、これらのいずれかによって構成される仮想的な計算機 (仮想マシン) である。DBサーバ 401 は、ネットワークアダプタ 413、メモリ 416、ローカル記憶デバイス 415 及びそれらに接続されたプロセッサ (典型的にはマイクロプロセッサ) 414 を有する。プロセッサ 414 は、コンピュータプログラム、例えば、OS (Operating System) 415 と、DBMS 412 と、DBMS 412 にクエリを発行する AP (Application Program) 411 とを実行する。メモリ 416 は、プロセッサ 414 によって実行されるプログラムと、プログラムが使用するデータとを一時的に記憶する。ローカル記憶デバイス 415 は、プログラム、及びプログラムが使用するデータを

10

20

30

40

50

格納する。ネットワークアダプタ 4 1 3 は、通信ネットワーク 4 0 3 と DB サーバ 4 0 1 とを接続する。AP 4 1 1 は、DB サーバ 4 0 1 ではなく、通信ネットワーク 4 0 3 に接続される図示しない別の計算機で動作しても良い。また、プロセッサ 4 1 4 は、ネットワークアダプタ 4 1 3 及びメモリ 4 1 6 等に接続された制御デバイスに含まれている要素で良い。制御デバイスは、プロセッサ 4 1 4 の他に、専用ハードウェア回路（例えば、データの暗号化及び / 又は復号化を行う回路）を含んで良い。

【 0 0 3 3 】

なお、DB サーバ 4 0 1 は、性能面や冗長性の観点から、プロセッサ 4 1 4、メモリ 4 1 6、ローカル記憶デバイス 4 1 5 及びネットワークアダプタ 4 1 3 のうちの少なくとも一つの要素を複数備えていても良い。また、DB サーバ 4 0 1 は、図示しない入力デバイス（例えば、キーボード及びポインティングデバイス）と表示デバイス（例えば液晶ディスプレイ）とを有して良い。入力デバイスと表示デバイスは一体になっていても良い。

10

【 0 0 3 4 】

DB サーバ 4 0 1 では、DBMS 4 1 2 が、AP 4 1 1 から発行されたクエリを実行する。DBMS 4 1 2 は、クエリを実行する際に、外部ストレージ装置 4 0 2 に格納された DB 4 5 1 に対する I / O 要求を OS 4 1 5 に発行する。OS 4 1 5 が、DBMS 4 1 2 から発行された I / O 要求を、外部ストレージ装置 4 0 2 に送信する。

【 0 0 3 5 】

外部ストレージ装置 4 0 2 は、本実施例では、ディスクアレイ装置のような、複数の記憶デバイス 4 4 3 を有する装置である。それに代えて、外部ストレージ装置 4 0 2 は、単一の記憶デバイスであっても良い。外部ストレージ装置 4 0 2 は、DB サーバ 4 0 1 が使用するデータ及びプログラムを記憶する。外部ストレージ装置 4 0 2 は、DB サーバ 4 0 1 から I / O 要求を受信し、I / O 要求に対応した処理を実行し、処理結果を DB サーバ 4 0 1 に送信する。

20

【 0 0 3 6 】

外部ストレージ装置 4 0 2 は、ネットワークアダプタ 4 4 1、記憶デバイス群 4 4 3 及びそれらに接続されたコントローラ 4 4 2 を有する。

【 0 0 3 7 】

ネットワークアダプタ 4 4 1 は、外部ストレージ装置 4 0 2 を通信ネットワーク 4 0 3 に接続する。

30

【 0 0 3 8 】

記憶デバイス群 4 4 3 は、1 以上の記憶デバイスを含む。記憶デバイスは、不揮発性の記憶媒体であって、例えば、磁気ディスク、フラッシュメモリ、その他半導体メモリである。記憶デバイス群 4 4 3 は、RAID (Redundant ARRAY of Independent Disks) に従い所定の RAID レベルでデータを記憶するグループであっても良い。記憶デバイス群 4 4 3 の記憶空間に基づいて、論理的な記憶デバイス（論理ボリューム）が DB サーバ 4 0 1 に提供されても良い。記憶デバイス群 4 4 3 は、DB 4 5 1 を記憶する。

【 0 0 3 9 】

コントローラ 4 4 2 は、例えば、メモリ及びプロセッサを含む。コントローラ 4 4 2 は、DB サーバ 4 0 1 からの I / O 要求に従って、DB 4 5 1 を記憶した記憶デバイス群 4 4 3 にデータを入出力する。例えば、コントローラ 4 4 2 は、DB サーバ 4 0 1 からの書込み要求に従う書込み対象のデータを記憶デバイス群 4 4 3 に格納する。また、DB サーバ 4 0 1 からの読出し要求に従う読出し対象のデータを記憶デバイス群 4 4 3 から読み出し、そのデータを DB サーバ 4 0 1 に送信する。

40

【 0 0 4 0 】

なお、外部ストレージ装置 4 0 2 は、性能面や冗長性確保の観点から、コントローラ 4 4 2 等の要素を複数備えても良い。

【 0 0 4 1 】

DBMS 4 1 2 は、業務データを含む DB 4 5 1 を管理する。DB 4 5 1 は、1 以上の表 4 6 2 や索引 4 6 1 を含む。表は 1 以上のレコードの集合であり、レコードは 1 以上の

50

カラムから構成される。索引は、表の中の1以上のカラムを対象に作成されるデータ構造であり、その索引が対象とするカラムを含む選択条件による表へのアクセスを高速化する。例えば、索引は、対象とするカラムの値毎にその値を含む表の中のレコードを特定する情報(Row ID)を保持するデータ構造であり、B木構造等が用いられる。DBの表の構成例や表同士の関連性の一例は、図1及び図2に示した通りである。

【0042】

DBMS 412は、クエリ受付部421、クエリ実行プラン生成部422、クエリ実行部423、実行タスク管理部426及びDBバッファ管理部427を含む。

【0043】

クエリ受付部421は、AP 421が発行するクエリを受け付ける。クエリは、例えばSQL (Structured Query Language) で記述される。

10

【0044】

クエリ実行プラン生成部422は、クエリ受付部421が受け付けたクエリについて、該クエリを実行するために必要な1以上のDBオペレーションを表す情報を有するクエリ実行プランを生成する。クエリ実行プランは、例えば、クエリの実行の際に行うべきDBオペレーションについてその実行順序を木構造で定義した情報であり、メモリ416に格納される。また、クエリ実行プランは、DBオペレーション毎に、タスクによる実行の際に要するメモリ資源の量を表す情報を含んで良い。例えば、或る表のレコードを読み出すDBオペレーションを表す情報を含むクエリ実行プランの場合、そのクエリ実行プランは、その表のレコードをメモリ416内に格納するためのメモリ資源量に関する情報を含んで良い。クエリ実行プランの一例は、図5及び図6に示した通りである。

20

【0045】

実行タスク管理部426は、クエリを実行するためのタスクを管理する。例えば、実行タスク管理部426は、タスクを生成するためのメモリ資源を管理しており、クエリ実行部423からのタスクの生成要求に従って、タスク自体に要するメモリ資源を確保してタスクを生成する。また、クエリ実行部423からのタスクの終了要求に従って、タスク自体に要するメモリ資源を解放してタスクを終了する。ここで、タスクとしては、任意のモジュールを採用することができる。例えば、タスクは、OS 415が管理するプロセス又はスレッドでも良いし、DBMS 412で実装される疑似プロセス又は疑似スレッドでも良い。

30

【0046】

DBバッファ管理部427は、DB 451内のデータを一時的に格納するための記憶領域(DBバッファ)を管理する。DBバッファは、メモリ416上に構築される。また、DBバッファは、ローカル記憶デバイス415上に構築されても良い。

【0047】

クエリ実行部423は、クエリ実行プラン生成部422が生成したクエリ実行プランに従って、クエリ実行プランに含まれる情報であるDBオペレーションを実行するためのタスクを動的に生成して実行する。そして、実行した結果をクエリ発行元に返す。例えば、クエリ実行部423は、(a) DBオペレーションを実行するためのタスクを実行タスク管理部426に要求して生成すること、(b) 生成されたタスクを実行することで、そのタスクに対応したDBオペレーションに必要なデータを読み出す(DB 451を有する外部ストレージ装置402へOS 415を介して読み出し要求を発行すること、(c) 上記(b)で実行されたタスクに対応したN番目のDBオペレーションの実行結果に基づき(N+1)番目のDBオペレーションを実行する場合には、その実行結果に基づくタスクを実行タスク管理部426に要求して新たに生成すること(Nは1以上の整数)、及び、(d) その新たに生成したタスクについて上記(b)及び上記(c)を行うこと、を行う。クエリ実行部423は、上記(b)及び(d)において、2以上の実行可能なタスクが存在する場合には、それら2以上のタスクのうち少なくとも二つのタスクを並行して実行する。

40

【0048】

50

クエリ実行部 4 2 3 は、タスクを新たに生成する場合に同時に生成可能なタスクの数である同時生成タスク数を決定する同時生成タスク数決定部 4 3 1 と、クエリの実行に割り当て可能なメモリ資源を管理するメモリ資源管理部 4 2 5 と、クエリの実行に関する情報を有するクエリ実行管理表 4 2 4 とを有する。

【 0 0 4 9 】

前述したクエリ受付部 4 2 1、クエリ実行プラン生成部 4 2 2、実行タスク管理部 4 2 6、DBバッファ管理部 4 2 7、クエリ実行部 4 2 3（同時生成タスク数決定部 4 3 1 及びメモリ資源管理部 4 2 5）の処理は、プロセッサ 4 1 4 が DBMS 4 1 2 を実行することによって実現される。上記の処理のうち、少なくとも一つの処理部に関する処理の中の少なくとも一部が、ハードウェアで行われても良い。また、本実施例の説明において、処理部が主語になる場合は、実際にはプロセッサ 4 1 4 によってその処理部を実行することを意味する。ここで、処理部の少なくとも一部がハードウェアで実現されている場合は、プロセッサ 4 1 4 に加えて、そのハードウェアも主語とされ得る。DBMS 4 1 2 等のコンピュータプログラムは、例えば、プログラムソースから DBサーバ 4 0 1 にインストールされて良い。プログラムソースは、例えば、DBサーバ 4 0 1 が読み取り可能な記憶メディアで良い。

【 0 0 5 0 】

また、図 1 0 に示した DBMS 4 1 2 の構成は、一例である。例えば、或る処理部が複数の処理部に分割されたり、複数の処理部の機能を統合した一つの処理部が構築されたりしても良い。

【 0 0 5 1 】

図 1 1 は、クエリ実行管理表 4 2 4 の構成を示す。

【 0 0 5 2 】

クエリ実行管理表 4 2 4 は、部分クエリ単位でその実行に関する情報を有する。「部分クエリ」とは、一つのタスクにより独立に実行を開始できる実行単位である。例えば、クエリ実行プランが複数の DB オペレーションを表す情報を含む場合、部分クエリは、一つのタスクにより独立に実行を開始できさえすれば、それら複数の DB オペレーションのうち 1 以上の DB オペレーションであっても良いし、それら複数の DB オペレーションのうちのある DB オペレーションの一部であっても良い。より具体的な一例として、サブクエリ内に SUM 等の集計処理を含むクエリの場合、そのクエリ実行プランは、サブクエリに関する部分クエリとクエリの残り部分（サブクエリの外側部分）に関する部分クエリとの二つの部分クエリを有する。クエリ実行部 4 2 3 は、クエリ実行プラン内の複数の部分クエリを並行して実行することができる。なお、図 5 及び図 6 のクエリ実行プランは、それぞれ一つの部分クエリからなる。

クエリ実行管理表 4 2 4 は、部分クエリ毎の情報として、優先度 9 0 1、最優先フラグ 9 0 2、目標実行時間 9 0 3、実行開始時刻 9 0 4、実行達成率 9 0 5、既割り当てメモリ資源量 9 0 6 及び既予約メモリ資源量 9 0 7 を有する。更にクエリ実行管理表 4 2 4 は、複数の部分クエリに共通の情報として、未予約メモリ資源量 9 0 8 及び割り当て可能総メモリ資源量 9 0 9 を有する。各種情報は、下記の通りである。

(*) 優先度 9 0 1 は、部分クエリを実行する優先度を表す。優先度 9 0 1 としての数値が高いほど、優先度が高い。優先度は、受け付けるクエリ自体に指定されていても良いし、クエリとは別にユーザから指定されても良い。優先度は、例えば、クエリ発行元の AP 4 1 1 に関する属性、クエリを発行したユーザに関する属性、及び / 又は、クエリ実行プランの内容を基に、DBMS 4 1 2 或いはユーザにより決定されて良い。

(*) 最優先フラグ 9 0 2 は、最優先して実行される部分クエリである（オン）か否かを示すフラグである。本フラグ 9 0 2 は、DBMS 4 1 2 が受け付けるクエリ自体に指定されても良いし、クエリとは別に DBMS 4 1 2 の外側からユーザにより手動で設定されても良いし、DBMS 4 1 2 の内部でクエリ実行プラン生成部 4 2 2 によりクエリ実行プランを生成される際にその内容に応じて設定されても良い。

(*) 目標実行時間 9 0 3 は、部分クエリの実行にかかる時間の目標値を示す。本目標値

10

20

30

40

50

は、ユーザにより指定されても良いし、DBMS 4 1 2（例えばクエリ実行プラン生成部 4 2 3）により決定されても良い。例えば、クエリ実行プラン生成部 4 2 3 は、コスト計算を基にクエリ実行プランを生成する際に、その計算に基づく予測実行時間を目標実行時間 9 0 3 と決めても良い。

（＊）実行開始時刻 9 0 4 は、部分クエリの実行を開始した時刻を示す。

（＊）実行達成率 9 0 5 は、部分クエリの実行が全体のうちのどこまで終了したかを示す。すなわち、実行達成率 9 0 5 は、部分クエリの実行の進捗を表す。例えば、実行達成率 9 0 5 は、上記のコスト計算に基づく予測全 I / O 数（予測 DB オペレーション数）に対する、実行を完了した I / O 数（完了 DB オペレーション数）の比率で良い。

（＊）既割り当てメモリ資源量 9 0 6 は、部分クエリの既予約メモリ資源量 9 0 7 のうち、その部分クエリの実行に割り当て済みのメモリ資源量を示す。

（＊）既予約メモリ資源量 9 0 7 は、部分クエリを実行するために予約済みのメモリ資源量を示す。

（＊）未予約メモリ資源量 9 0 8 は、割り当て可能総メモリ資源量 9 0 9 のうちいずれの部分クエリの実行にも予約されていないメモリ資源の総量である。

（＊）割り当て可能総メモリ資源量 9 0 9 は、全ての部分クエリの実行に割り当て可能なメモリ資源の総量である。このメモリ資源量 9 0 9 の値が、割り当て上限値に相当する。

【 0 0 5 3 】

メモリ資源管理部 4 2 5 は、クエリの実行に用いるメモリ資源に関する情報 9 0 6 ~ 9 0 9 を管理する。メモリ資源管理部 4 2 5 は、少なくとも一つの部分クエリの実行を開始する前に、割り当て可能総メモリ資源量 9 0 9 を登録していても良い。なお、クエリ実行部 4 2 3 は、クエリ実行管理表 4 2 4 を排他的に更新して良い。

【 0 0 5 4 】

以下、図 1 2 ~ 図 2 5 を参照して、本実施例で行われる種々の処理を説明する。

【 0 0 5 5 】

図 1 2 は、DBMS 4 1 2 が受け付けたクエリの実行に際して、タスクを生成し実行する処理の全体の流れを示す。図 1 2 は一つのクエリ実行の流れを示す。なお、同時に実行するクエリが 2 以上ある場合には、DBMS 4 1 2 は、該 2 以上のクエリについて並行して図 1 2 に示す処理を行って良い。

【 0 0 5 6 】

S 1 0 0 1 で、クエリ実行プラン生成部 4 2 2 は、DB 4 5 1 のデータの統計情報（以下、「DB 統計情報」）に基づいて、クエリ受付部 4 2 1 が受け付けたクエリに関するクエリ実行プランを生成する。クエリ実行プラン生成部 4 2 2 は、一つ以上のクエリ実行プランの候補の各々についてコスト計算を行い、そのうち一つのクエリ実行プランの候補をクエリ実行プランとして選んでも良い。DB 統計情報は、例えば、各表 4 6 2 について、レコードの数を表す情報、カラム毎のデータの最大値及び最小値に関する情報、或いはカラム毎のデータの分布状況を表すヒストグラム等の統計情報を含んで良い。DB 統計情報は、ローカル記憶デバイス 4 1 5 又はメモリ 4 1 6 に格納される。DB 統計情報は、DBMS 4 1 2 が DB 4 5 1 をチェックすることにより生成されても良いし、DBMS 4 1 2 の外部から入力されても良い。また、クエリ実行プランは、DB 統計情報のみならず、クエリの記述に基づいて生成されても良い。クエリ実行プラン生成部 4 2 2 は、S 1 0 0 1 を行った時刻を、該当クエリに対応する部分クエリの実行開始時刻 9 0 4 としてクエリ実行管理表 4 2 4 に登録するようクエリ実行部 4 2 3 に指示して良い。また、クエリ実行プラン生成部 4 2 2 は、生成するクエリ実行プランに基づいて、クエリ実行管理表 4 2 4 における、該当クエリに対応する部分クエリの優先度 9 0 1、最優先フラグ 9 0 2、及び、目標実行時間 9 0 3 を登録するようクエリ実行部 4 2 3 に指示しても良い。

【 0 0 5 7 】

S 1 0 0 2 で、クエリ実行部 4 2 3 は、S 1 0 0 1 で生成したクエリ実行プランに基づいて、新たに生成可能なタスク数である生成可能タスク数を算出する。例えば、クエリ実行部 4 2 3 は、クエリ実行プランを基に、複数のタスクにより並行して実行可能な部分ク

10

20

30

40

50

エリの数を判別し、その判別した部分クエリの数を生成可能タスク数として算出する。

【 0 0 5 8 】

S 1 0 0 3 で、クエリ実行部 4 2 3 (同時生成タスク数決定部 4 3 1) は、同時生成タスク数を決定する処理である同時生成タスク数決定処理を行う。同時生成タスク数決定処理の詳細は、図 1 5 に示す通りである。図 1 5 によれば、クエリ実行部 4 2 3 は、同時生成タスク数決定処理により、同時生成タスク数のタスク各々に割り当てるためのメモリ資源を予約する。

【 0 0 5 9 】

S 1 0 0 4 で、クエリ実行部 4 2 3 は、S 1 0 0 3 で決定された同時生成タスク数と同じ数のタスクの生成を実行タスク管理部 4 2 6 に要求する。実行タスク管理部 4 2 6 は、要求された数のタスクを生成する。なお、生成したタスクの数が同時生成タスク数を下回る場合には、クエリ実行部 4 2 3 は、S 1 0 0 3 で予約したメモリ資源のうち、同時生成タスク数の中で生成していないタスクに関するメモリ資源の予約を解消しても良い。

【 0 0 6 0 】

S 1 0 0 5 で、クエリ実行部 4 2 3 は、S 1 0 0 4 で生成した各々のタスクを実行する処理 (タスク実行処理) を並行して行う。実行されるタスク実行処理の詳細は、図 1 3 に示す通りである。なお、クエリ実行部 4 2 3 は、S 1 0 0 3 で決定した同時生成タスク数がゼロである場合には、S 1 0 0 4 及び S 1 0 0 5 のステップでは何も行わない。

【 0 0 6 1 】

S 1 0 0 6 で、クエリ実行部 4 2 3 は、S 1 0 0 2 で算出した生成可能タスク数のタスクが全て生成されたか否かを判断する。クエリ実行部 4 2 3 は、該判断の結果が肯定的であれば、処理を終了する。該判断の結果が否定的であれば、S 1 0 0 7 を行う。なお、クエリ実行部 4 2 3 は、クエリの実行結果を、S 1 0 0 5 で実行したタスクの処理の延長で生成する。

【 0 0 6 2 】

S 1 0 0 7 で、クエリ実行部 4 2 3 は、新たなタスクの生成が可能か否かを判断する。例えば、クエリ実行部 4 2 3 は、新たなタスクの生成が可能か否かを実行タスク管理部 4 2 6 に問い合わせる。実行タスク管理部 4 2 6 は、例えば、タスクを生成するために管理しているメモリ資源の割り当て状況に応じて、問合せに回答する。また、クエリ実行部 4 2 3 は、例えば、クエリ実行管理表 4 2 4 を参照し、未予約メモリ資源量 9 0 8 が新たなタスクに対応する DB オペレーションに必要なメモリ資源量を上回るかを判定する。クエリ実行部 4 2 3 は、該判定結果と実行タスク管理部 4 2 6 からの上記回答とに基づき、新たなタスクの生成が可能か否かを判断する。クエリ実行部 4 2 3 は、S 1 0 0 7 の判断の結果が肯定的であれば、再び S 1 0 0 3 を行い、S 1 0 0 7 の判断の結果が否定的であれば、S 1 0 0 8 を行う。

【 0 0 6 3 】

S 1 0 0 8 で、クエリ実行部 4 2 3 は、タスク生成保留処理を行う。タスク生成保留処理の詳細は、図 1 4 に示す通りである。

【 0 0 6 4 】

図 1 3 は、タスク実行処理 (図 1 2 の S 1 0 0 5 及び図 1 3 の S 1 1 0 6) の流れを示す。図 1 3 は、一つの親タスク (図 1 3 の説明において「対象親タスク」と言う) を例に取った流れを示す。また、図 1 3 の説明において、「親タスク」とは、任意の一つのタスクを意味する。親タスクを元に生成されたタスクを「子タスク」と言う。クエリ実行部 4 2 3 により子タスクを元に別のタスクが生成された場合、子タスクが親タスクに相当し、生成された別のタスクが子タスクに相当する。

【 0 0 6 5 】

S 1 1 0 1 で、クエリ実行部 4 2 3 (メモリ資源管理部 4 2 5) は、対象親タスクへのメモリ資源割り当て処理を行う。メモリ資源割り当て処理の詳細は、図 1 6 に示す通りである。

【 0 0 6 6 】

10

20

30

40

50

S 1 1 0 2 で、クエリ実行部 4 2 3 は、対象親タスクに割り当てたメモリ資源に、対象親タスクに対応する DB オペレーションを実行するための情報（例えば、DB オペレーションに要するデータのアクセス先、及び、結果を生成するために必要なフェッチ済みのデータ）を設定する。そして、対象親タスクに対応する DB オペレーションを実行する。対象親タスクに対応する DB オペレーションを実行するための情報は、対象親タスクの生成元であるタスクが有する情報に基づいて設定される。クエリ実行部 4 2 3 は、対象親タスクを実行することで、例えば、DB 4 5 1 からデータを読み出す要求を発行し、読み出されたデータに対して DB オペレーションを実行する。なお、S 1 1 0 2 を終了したときに、クエリ実行部 4 2 3 は、実行した部分クエリに対応する実行達成率 9 0 5（クエリ実行管理表 4 2 4 における一つの値）を更新して良い。

10

【 0 0 6 7 】

S 1 1 0 3 で、クエリ実行部 4 2 3 は、S 1 1 0 2 で実行した DB オペレーションの実行結果を基に、クエリ実行プラン上の次の DB オペレーションを行うために新たに生成可能な子タスクの数である生成可能タスク数を算出する。例えば、S 1 1 0 2 で、クエリ実行部 4 2 3 は、対象親タスクにより、図 1 で示した索引 A の Row ID List 「a 1」を検索する DB オペレーションを実行したとする。このとき、クエリ実行部 4 2 3 は、該実行結果に基づき、表 A の 1 レコード目を含むデータと 3 レコード目を含むデータをそれぞれフェッチするため（図 5 のクエリ実行プラン上の次の DB オペレーションのため）に新たに生成できる子タスクの数を 2 と判別し、該数を生成可能タスク数として算出する。なお、対象親タスクに対応する DB オペレーションがクエリ実行の結果生成等であり次の DB オペレーションが無い場合には、クエリ実行部 4 2 3 は、生成可能タスク数をゼロと算出する。生成可能タスク数をゼロと算出した場合、クエリ実行部 4 2 3 は、S 1 1 0 4、S 1 1 0 5、S 1 1 0 6、及び、S 1 1 0 7 のステップでは何も行わず、S 1 1 0 8 のステップを行う。

20

【 0 0 6 8 】

S 1 1 0 4 で、クエリ実行部 4 2 3（同時生成タスク数決定部 4 3 1）は、同時生成タスク数決定処理を行う。同時生成タスク数決定処理の詳細は、図 1 5 に示す通りである。図 1 5 によれば、クエリ実行部 4 2 3 は、同時生成タスク数決定処理により、同時生成タスク数のタスク各々に割り当てるためのメモリ資源を予約する。

30

【 0 0 6 9 】

S 1 1 0 5 で、クエリ実行部 4 2 3 は、S 1 1 0 4 で決定された同時生成タスク数と同じ数の子タスクの生成を実行タスク管理部 4 2 6 に要求する。実行タスク管理部 4 2 6 は、要求された数のタスクを生成する。なお、生成した子タスクの数が同時生成タスク数を下回る場合には、クエリ実行部 4 2 3 は、S 1 1 0 4 で予約したメモリ資源のうち、同時生成タスク数の中で生成していないタスクに関するメモリ資源の予約を解消しても良い。

【 0 0 7 0 】

S 1 1 0 6 で、クエリ実行部 4 2 3 は、S 1 1 0 5 で生成した各々の子タスクについて、S 1 1 0 2 での DB オペレーションの実行結果に基づき、次の DB オペレーションに対応させてタスク実行処理（図 1 3 に示す処理）を並行して行う。

40

【 0 0 7 1 】

S 1 1 0 7 で、クエリ実行部 4 2 3 は、S 1 1 0 3 で算出した生成可能タスク数の子タスクが全て生成された否かを判断する。クエリ実行部 4 2 3 は、該判断の結果が肯定的であれば、S 1 1 0 8 を行い、該判断の結果が否定的であれば、S 1 1 0 9 を行う。

【 0 0 7 2 】

S 1 1 0 8 で、クエリ実行部 4 2 3（メモリ資源管理部 4 2 5）は、実行した対象親タスクについて解放可能なメモリ資源を解放するメモリ資源解放処理を行う。メモリ資源解放処理の詳細は、図 1 7 に示す通りである。S 1 1 0 8 の後、クエリ実行部 4 2 3 は、対象親タスクの終了を実行タスク管理部 4 2 6 に指示する。

【 0 0 7 3 】

S 1 1 0 9 で、クエリ実行部 4 2 3 は、新たな子タスクの生成が可能か否かを判断する

50

。この処理は、S 1 0 0 7と同様である。クエリ実行部 4 2 3 は、S 1 1 0 9の判断の結果が肯定的であれば、再びS 1 1 0 4を行い、S 1 1 0 9の判断の結果が否定的であれば、S 1 1 1 0を行う。

【 0 0 7 4 】

S 1 1 1 0で、クエリ実行部 4 2 3 は、タスク生成保留処理を行う。タスク生成保留処理の詳細は、図 1 4 に示す通りである。

【 0 0 7 5 】

図 1 4 は、タスク生成保留処理（図 1 2 の S 1 0 0 8、及び、図 1 3 の S 1 1 1 0）の流れを示す。図 1 4 は、一つのタスクを例にとった流れを示す。

【 0 0 7 6 】

S 1 2 0 1で、クエリ実行部 4 2 3 は、新たなタスクの生成が可能になるか、一定時間が経過するまで待つ。「新たなタスクの生成が可能か否か」を判断する処理は、S 1 0 0 7と同様である。S 1 2 0 1では、クエリ実行部 4 2 3 は、例えば、新たなタスクの生成が可能になるまでS 1 0 0 7と同様の処理を所定の時間間隔で繰り返す。タスク生成保留処理において、例えば、下記（x）又は（y）、

（x）実行している部分クエリについて、優先度 9 0 1が高くなったり、最優先フラグ 9 0 2が付与されたりする等の原因により、該部分クエリの実行に割り当て可能なメモリ資源量が増えた、

（y）未予約メモリ資源量 9 0 8又は割り当て可能総メモリ資源量 9 0 9が増えた、等の事象が発生することにより、新たなタスクを生成可能になることが考えられる。なお、S 1 2 0 1で、クエリ実行部 4 2 3 は、クエリ実行管理表 4 2 4を参照して、実行している部分クエリに対応する実行開始時刻 9 0 4から所定の時間を経過した場合、この部分クエリを含むクエリの実行をエラー終了しても良い。クエリの実行をエラー終了する際には、クエリ実行部 4 2 3 は、該クエリの実行に関連するメモリ資源を解放且つ予約解消して、関連するタスクを終了する。

【 0 0 7 7 】

図 1 5 は、同時生成タスク数決定処理（図 1 2 の S 1 0 0 3、及び、図 1 3 の S 1 1 0 4）の流れを示す。

【 0 0 7 8 】

S 1 3 0 1で、同時生成タスク数決定部 4 3 1 は、新たに生成するタスク当たり割り当てを要するメモリ資源量である第 1 のメモリ資源量を、クエリ実行プランに含まれる情報である、新たに生成するタスクに対応する DB オペレーションに要するメモリ資源量に基づき算出する。なお、DB オペレーションに必要な一部のメモリ資源を、新たに生成するタスクを含む複数のタスクの間で共有する場合には、同時生成タスク数決定部 4 3 1 は、該共有するメモリ資源の量をタスク間で重複して計上しないように、第 1 のメモリ資源量を算出して良い。

【 0 0 7 9 】

S 1 3 0 2で、同時生成タスク数決定部 4 3 1 は、S 1 3 0 1で算出した第 1 のメモリ資源量と、本同時生成タスク数決定処理の直前のステップ（図 1 2 の S 1 0 0 2、又は、図 1 3 の S 1 1 0 3）で算出した生成可能タスク数とを基に、新たに生成するタスクに割り当てるために予約するメモリ資源の量である予約メモリ資源量を決定する。予約メモリ資源量は、例えば、第 1 のメモリ資源量と生成可能タスク数との積で良い。

【 0 0 8 0 】

S 1 3 0 3で、同時生成タスク数決定部 4 3 1 は、S 1 3 0 2で決定された予約メモリ資源量分のメモリ資源の予約をメモリ資源管理部 4 2 5に要求する。メモリ資源管理部 4 2 5は、その要求を受けて、図 1 8 に示すメモリ資源予約処理を行い、予約に成功したメモリ資源の量を同時生成タスク数決定部 4 3 1に通知する。予約に成功するメモリ資源の量は、例えば、第 1 のメモリ資源量の整数倍で良い。

【 0 0 8 1 】

S 1 3 0 4で、同時生成タスク数決定部 4 3 1 は、S 1 3 0 3で予約に成功したメモリ

10

20

30

40

50

資源の量と、S 1 3 0 1で算出した第1のメモリ資源量とを基に、同時生成タスク数を決定する。同時生成タスク数は、例えば、(S 1 3 0 3で予約に成功したメモリ資源の量)を(第1のメモリ資源量)で除した値が良い。同時生成タスク数決定部4 3 1は、余りのメモリ資源が生じた場合、メモリ資源管理部4 2 5に余りのメモリ資源量分の予約を解消するよう要求しても良い。

【0082】

図16は、メモリ資源割り当て処理(図13のS 1 1 0 1)の流れを示す。図16は、一つのタスク(図16の説明において「対象タスク」と言う)を例に取った流れを示す。なお、図16の説明において、対象タスクにより実行する部分クエリを、「対象部分クエリ」と言う。

10

【0083】

S 1 4 0 1で、メモリ資源管理部4 2 5は、クエリ実行管理表4 2 4を参照し、対象部分クエリの情報を得る。

【0084】

S 1 4 0 2で、メモリ資源管理部4 2 5は、 $P + Q$ が R 以下であるか否かを判断する。 P 、 Q 及び R は、下記の通りである。また、下記において、新規割り当てメモリ資源量 Q は、クエリ実行プランに基づく第1のメモリ資源量(図15のS 1 3 0 1での算出値)から予測することができる。しかし、該予測は必ずしも完全というわけではなく、実際に対象タスクを実行する際の Q は、予測したメモリ資源量、すなわち図15のS 1 3 0 3で予約したメモリ資源の量とは異なる可能性もある。

20

P = 対象部分クエリに対応する既割り当てメモリ資源量9 0 6。

Q = 対象タスクを実行するために必要とする新規割り当てメモリ資源量。

R = 対象部分クエリに対応する既予約メモリ資源量9 0 7。

【0085】

S 1 4 0 2の判断の結果が否定的であれば(S 1 4 0 2 : F)、メモリ資源管理部4 2 5は、対象部分クエリに対応する下記(1)~(3)のイベント、

(1) 既割り当てメモリ資源量9 0 6が減少する、

(2) 既予約メモリ資源量9 0 7が増加する、及び

(3) 一定時間が経過する、

のうちのいずれかのイベントを検出するまで待つ。そのいずれかのイベントを検出した場合に、メモリ資源管理部4 2 5は、再度、S 1 4 0 2を行って良い。S 1 4 0 2の判断の結果が肯定的であれば(S 1 4 0 2 : T)、メモリ資源管理部4 2 5は、S 1 4 0 3を行う。

30

【0086】

S 1 4 0 3で、メモリ資源管理部4 2 5は、クエリ実行管理表4 2 4を更新する。例えば、S 1 4 0 2での新規割り当てメモリ資源量を、対象部分クエリに対応する既割り当てメモリ資源量9 0 6に加算する。

S 1 4 0 4で、メモリ資源管理部4 2 5は、予約しているメモリ資源の中から新規割り当てメモリ資源量分のメモリ資源を、対象タスクに割り当てる。

【0087】

図17は、メモリ資源解放処理(図13のS 1 1 0 8)の流れを示す。図17は、一つのタスク(図17の説明において「対象タスク」と言う)を例に取った流れを示す。なお、図17の説明において、対象タスクにより実行した部分クエリを、「対象の部分クエリ」と言う。

40

【0088】

S 1 5 0 1で、メモリ資源管理部4 2 5は、対象タスクに対して割り当てられているメモリ資源を解放する。なお、メモリ資源管理部4 2 5は、対象タスクに割り当てられたメモリ資源のうち、対象タスクと異なるタスクにより共有されるメモリ資源については解放しない。

【0089】

50

S 1 5 0 2 で、メモリ資源管理部 4 2 5 は、S 1 5 0 1 で解放したメモリ資源量分のメモリ資源の予約を解消する。

【 0 0 9 0 】

S 1 5 0 3 で、メモリ資源管理部 4 2 5 は、対象部分クエリに対応するクエリ実行管理表 4 2 4 の情報を更新する。例えば、メモリ資源管理部 4 2 5 は、S 1 5 0 1 で解放されたメモリ資源の量を、対象部分クエリに対応する既割り当てメモリ資源量 9 0 6 及び既予約メモリ資源量 9 0 7 から減算し、且つ、当該メモリ資源の量を未予約メモリ資源量 9 0 8 に加算する。

【 0 0 9 1 】

図 1 8 は、メモリ資源予約処理（図 1 5 の S 1 3 0 3 ）の流れを示す。図 1 8 は、一つのタスク（図 1 8 の説明において「対象タスク」と言う）を例に取った流れを示す。なお、図 1 8 の説明において、対象タスクに関する部分クエリを、「対象部分クエリ」と言う。

10

【 0 0 9 2 】

S 1 6 0 1 で、メモリ資源管理部 4 2 5 は、クエリ実行管理表 4 2 4 を参照し、対象部分クエリの情報を得る。

【 0 0 9 3 】

S 1 6 0 2 で、メモリ資源管理部 4 2 5 は、対象部分クエリに対応する最優先フラグ 9 0 2 がオンか否かを判断する。

【 0 0 9 4 】

20

S 1 6 0 2 の判断の結果が肯定的であれば（S 1 6 0 2 : T）、S 1 6 0 3 で、メモリ資源管理部 4 2 5 は、下記（a）及び（b）、

（a）要求された予約メモリ資源量、

（b）未予約メモリ資源量 9 0 8、

のうち小さい方のメモリ資源量を算出する。

【 0 0 9 5 】

S 1 6 0 2 の判断の結果が否定的であれば（S 1 6 0 2 : F）、S 1 6 0 4 で、メモリ資源管理部 4 2 5 は、下記（a）～（c）、

（a）要求された予約メモリ資源量、

（b）未予約メモリ資源量 9 0 8、

（c）対象部分クエリの実行について割り当て可能なメモリ資源量の上限値から、対象部分クエリに対応する既予約メモリ資源量 9 0 7 を減算することにより得られるメモリ資源量、

30

のうちの最小のメモリ資源量を算出する。

【 0 0 9 6 】

なお、（c）の「対象部分クエリの実行について割り当て可能なメモリ資源量の上限値」とは、割り当て可能総メモリ資源量 9 0 9 を、実行中の部分クエリ毎の優先度 9 0 1 に基づき、部分クエリ毎に配分（例えば比例配分）した値のうちの、対象部分クエリに対応した値である。例えば、図 1 1 に示したように、割り当て可能総メモリ資源量 9 0 9 が「1 0 0 MB（メガバイト）」であり、部分クエリ 1～3 の優先度が「3」、「2」及び「5」である場合を考える。また、割り当て可能総メモリ資源量 9 0 9 を部分クエリ毎の優先度 9 0 1 に基づき比例配分すると仮定する。このとき、各部分クエリの実行について割り当て可能なメモリ資源量の上限値は、部分クエリ 1 については「3 0 MB」、部分クエリ 2 については「2 0 MB」、部分クエリ 3 については「5 0 MB」である。

40

【 0 0 9 7 】

また、例えば、同時に実行している部分クエリの数が増えた場合や、最優先フラグ 9 0 2 がオンである部分クエリを実行中の場合、（c）のメモリ資源量は、負の値になることがある。その場合、メモリ資源管理部 4 2 5 は、S 1 6 0 4 で算出するメモリ資源量を、ゼロとする。

【 0 0 9 8 】

50

S 1 6 0 5 で、メモリ資源管理部 4 2 5 は、クエリ実行管理表 4 2 4 を更新する。例えば、メモリ資源管理部 4 2 5 は、対象部分クエリに対応する既予約メモリ資源量 9 0 7 に、S 1 6 0 3 又は S 1 6 0 4 で算出したメモリ資源量を加算する。更に、メモリ資源管理部 4 2 5 は、未予約メモリ資源量 9 0 8 から、S 1 6 0 3 又は S 1 6 0 4 で算出したメモリ資源量を減算する。

S 1 6 0 6 で、メモリ資源管理部 4 2 5 は、クエリ実行に割り当て可能なメモリ資源のうち、予約されていないメモリ資源の中から、S 1 6 0 3 又は S 1 6 0 4 で算出した量のメモリ資源を予約する。

【 0 0 9 9 】

本実施例で述べたクエリの実行に割り当て可能なメモリ資源の総量は、様々な要因により増減することがある。例えば、DBMS 4 1 2 の内部において、実行タスク管理部 4 2 6 で管理するメモリ資源量（クエリを実行するためのタスク用途）を増減すること、又は、DBバッファ管理部 4 2 7 で管理するメモリ資源量（DBバッファ用途）を増減すること、等が起因する。また、例えば、DBMS 4 1 2 の外部において、同じメモリ 4 1 6 上で実行される他のコンピュータプログラムのメモリ資源の利用状況、メモリ 4 1 6 の増設又は減設、又は、DBサーバ 4 0 1 が仮想マシンである場合における該仮想マシンの総メモリ資源量の変化、等に起因する。特に、近年普及しつつあるクラウド環境における計算機資源の仮想化により、クエリの実行に割り当て可能なメモリ資源の総量は、状況に応じて変化することがある。

図 1 9 は、メモリ資源増加処理の流れを示す。

【 0 1 0 0 】

メモリ資源増加処理は、クエリの実行に割り当て可能なメモリ資源の総量である割り当て可能総メモリ資源量 9 0 9 を増加するための処理である。この処理は、上記したメモリ資源量の増加要因が発生した際に開始されて良い。

【 0 1 0 1 】

S 1 7 0 1 で、メモリ資源管理部 4 2 5 は、クエリ実行管理表 4 2 4 を参照する。

【 0 1 0 2 】

S 1 7 0 2 で、メモリ資源管理部 4 2 5 は、クエリ実行に用いるために、割り当て可能総メモリ資源量 9 0 9 に増加する量のメモリ資源を初期化し、予約されていないメモリ資源として管理する。

【 0 1 0 3 】

S 1 7 0 3 で、メモリ資源管理部 4 2 5 は、クエリ実行管理表 4 2 4 を更新する。例えば、メモリ資源管理部 4 2 5 は、S 1 7 0 2 で増加されたメモリ資源の量を、未予約メモリ資源量 9 0 8 及び割り当て可能総メモリ資源量 9 0 9 に加算する。

【 0 1 0 4 】

図 2 0 は、メモリ資源減少処理の流れを示す。

【 0 1 0 5 】

メモリ資源減少処理は、割り当て可能総メモリ資源量 9 0 9 を減少するための処理である。この処理は、上記したメモリ資源量の減少要因が発生した際に開始されて良い。

【 0 1 0 6 】

S 1 8 0 1 で、メモリ資源管理部 4 2 5 は、クエリ実行管理表 4 2 4 を参照する。

【 0 1 0 7 】

S 1 8 0 2 で、メモリ資源管理部 4 2 5 は、未予約メモリ資源量 9 0 8 と、割り当て可能総メモリ資源量 9 0 9 から減少する量とのうち、小さい方の量を判別する。メモリ資源管理部 4 2 5 は、該判別した量のメモリ資源を、クエリ実行に割り当て可能なメモリ資源のうち、予約されていないメモリ資源の中から解放する。

【 0 1 0 8 】

S 1 8 0 3 で、メモリ資源管理部 4 2 5 は、クエリ実行管理表 4 2 4 を更新する。例えば、メモリ資源管理部 4 2 5 は、S 1 8 0 2 で解放されたメモリ資源の量を、未予約メモリ資源量 9 0 8 及び割り当て可能総メモリ資源量 9 0 9 から減少する。

10

20

30

40

50

【0109】

S1804で、メモリ資源管理部425は、割り当て可能総メモリ資源量909から減少する量のメモリ資源が解放されたか否かを判断する。該判断の結果が肯定的であれば、メモリ資源管理部425は、処理を終了する。一方、該判断の結果が否定的であれば、メモリ資源管理部425は、再びS1802を行う。該判断の結果が否定的となるケースは、S1802で、メモリ資源管理部425が未予約メモリ資源量908分のメモリ資源を解放したケースである。すなわち、S1802で解放されたメモリ資源の量は、割り当て可能総メモリ資源量909から減少する量よりも少ないケースである。該ケースにおいて、メモリ資源管理部425は、再度S1802を行い、割り当て可能総メモリ資源量909から減少する量と、前回のS1802で解放されたメモリ資源の量との差分の量のメモリ資源を解放する。

10

【0110】

図21は、サーバメモリ資源増加処理の流れを示す。

サーバメモリ資源増加処理は、例えば、DBサーバ401のメモリ416が増設された場合に、又は、DBサーバ401が仮想マシンであり該仮想マシンの総メモリ資源量が増加された場合に、OS415に認識される総メモリ資源量を増加する処理である。

【0111】

S1901で、OS415は、総メモリ資源量を増加する。例えば、DBサーバ401の管理者により、DBサーバ401のメモリ416を増設されたとする。又は、DBサーバ401が仮想マシンである場合には、該仮想マシンの総メモリ資源量が増加されたとする。このとき、OS415は、上記増設又は増加された量分の総メモリ資源量の増加を認識する。

20

【0112】

S1902で、DBMS412内のメモリ資源管理部425は、S1901で増加されたメモリ資源の量について、図19に示したメモリ資源増加処理を行う。なお、DBMS412は、DBMS412内で使用するメモリ資源である、実行タスク管理部426で管理するメモリ資源又はDBバッファ管理部427で管理するメモリ資源等の量を増加しても良い。

【0113】

図22は、サーバメモリ資源減少処理の流れを示す。

30

サーバメモリ資源減少処理は、例えば、DBサーバ401のメモリ416が減設された場合に、又は、DBサーバ401が仮想マシンであり該仮想マシンの総メモリ資源量が減少された場合に、OS415に認識される総メモリ資源量を減少する処理である。

【0114】

S2001で、DBMS412内のメモリ資源管理部425は、減少するメモリ資源量について、図20に示したメモリ資源減少処理を行う。なお、DBMS412は、DBMS412内で使用するメモリ資源である、実行タスク管理部426で管理するメモリ資源又はDBバッファ管理部427で管理するメモリ資源等の量を減少しても良い。

【0115】

S2002で、OS415は、総メモリ資源量を減少する。例えば、DBサーバ401の管理者により、S2001で減少されたメモリ資源の量に相当するメモリ416が減設されたとする。又は、DBサーバ401が仮想マシンである場合には、S2001で減少されたメモリ資源の量だけ該仮想マシンの総メモリ資源量が減少されたとする。このとき、OS415は、上記減設又は減少された量分の総メモリ資源量の減少を認識する。

40

【0116】

図23は、優先度手動変更処理の流れを示す。以下、一つの部分クエリ(図23の説明において「対象部分クエリ」と言う)を例に取る。

【0117】

S2101で、クエリ受付部421は、対象部分クエリに関する優先度変更指示を受け付け、クエリ実行部423に、対象部分クエリの優先度変更指示を出す。優先度変更指示

50

の送信元は、対象部分クエリに関するクエリの発行元と同じであっても異なっても良い。クエリ実行部 4 2 3 は、該指示を受けて、クエリ実行管理表 4 2 4 の対象部分クエリに対応する優先度 9 0 1 又は最優先フラグ 9 0 2 を、該指示の内容に従い変更する。

【 0 1 1 8 】

図 2 4 は、優先度自動変更処理の流れを示す。以下、優先度を変更する対象である一つの部分クエリ（図 2 3 の説明において「変更対象部分クエリ」と言う）を例に取る。優先度を変更する対象の部分クエリとは、例えば、優先度を自動的に変更することを指定されている部分クエリである。なお、本優先度自動変更処理は、優先度を変更する対象の部分クエリ各々に対して行われて良い。また、優先度自動変更処理は、例えば、一定の時間間隔で行われて良い。

10

【 0 1 1 9 】

S 2 2 0 1 で、クエリ実行部 4 2 3 は、クエリ実行管理表 4 2 4 を参照し、変更対象部分クエリの情報を得る。

【 0 1 2 0 】

S 2 2 0 2 で、クエリ実行部 4 2 3 は、変更対象部分クエリに対応する優先度を、変更対象部分クエリに対応する目標実行時間 9 0 3 と、実行経過時間（現在時刻から実行開始時刻 9 0 4 を減算することにより得られる値）と、実行達成率 9 0 5 とのうちの少なくとも一つを基に、新たに決定する。

【 0 1 2 1 】

S 2 2 0 3 で、クエリ実行部 4 2 3 は、クエリ実行管理表 4 2 4 内の変更対象部分クエリに対応する優先度 9 0 1 を、S 2 2 0 2 で決定した優先度に変更する。

20

【 0 1 2 2 】

なお、クエリ実行部 4 2 3 が変更対象部分クエリの優先度を決定する方法としては、例えば、下記（ 1 ）～（ 3 ）のうちの少なくとも一つの方法がある。

（ 1 ）上記経過時間が目標実行時間 9 0 3 に所定割合接近した場合に、優先度を以前の値から一つ上げる。

（ 2 ）上記経過時間が目標実行時間 9 0 3 を所定割合経過した場合に、優先度を以前の値から一つ上げる。

（ 3 ）実行達成率 9 0 5 が所定割合以上であり且つ上記経過時間が目標実行時間 9 0 3 の所定割合未満である場合に、優先度を以前の値から一つ下げる。

30

【 0 1 2 3 】

図 2 5 は、タスク追加生成処理の流れを示す。

【 0 1 2 4 】

タスク追加生成処理は、例えば、下記（ 1 ）又は（ 2 ）の事象が発生した契機に開始される処理である。

（ 1 ）クエリ実行管理表 4 2 4 の割り当て可能総メモリ資源量 9 0 9 又は未予約メモリ資源量 9 0 8 が増加した。

（ 2 ）クエリ実行管理表 4 2 4 のある部分クエリの優先度 9 0 1 が変化した、又は、最優先フラグ 9 0 2 がオンになった。

なお、本タスク追加生成処理は、クエリ実行管理表 4 2 4 の未予約メモリ資源量 9 0 8 が所定値以上の場合に、クエリ実行管理表 4 2 4 の全ての部分クエリについて、タスクの生成を保留するタスクが存在しなくなるまで繰り返されても良い。

40

【 0 1 2 5 】

S 2 3 0 1 で、クエリ実行部 4 2 3 は、クエリ実行管理表 4 2 4 を参照し、タスクの生成を保留しているタスク（図 1 2 の S 1 0 0 8 又は図 1 3 の S 1 1 1 0 ）が存在する部分クエリのうち、優先度 9 0 1 が最も高い部分クエリを選択する。

【 0 1 2 6 】

S 2 3 0 2 で、クエリ実行部 4 2 3 は、S 2 3 0 1 で選択した部分クエリにおいて、タスクの生成を待っているタスクのうち、実行優先度の高いタスクを選択する。例えば、クエリ実行部 4 2 3 は、タスクに対応する DB オペレーションより後続の DB オペレーショ

50

ンの数が最も少ないタスクを選択する。

【 0 1 2 7 】

S 2 3 0 3 で、クエリ実行部 4 2 3 は、S 2 3 0 2 で選択されたタスクについてタスクの生成待ちを優先的に解除する。

【実施例 2】

【 0 1 2 8 】

以下、実施例 2 を説明する。その際、実施例 1 との相違点を主に説明し、実施例 1 との共通点については説明を省略又は簡略する。

実施例 2 においても、DBMS 4 1 2 は、タスクを新たに生成する都度に、同時生成タスク数決定処理を行う。実施例 2 では、DBMS 4 1 2 は、同時生成タスク数決定処理において、新たに生成するタスク当たり割り当てを要するメモリ資源量である第 1 のメモリ資源量を、新たに生成するタスクに対応する、結果生成に至るまでの後続の DB オペレーションに要するメモリ資源量に基づき算出する（実施例 1 では、第 1 のメモリ資源量は、新たに生成するタスクに対応する一つの DB オペレーションに要するメモリ資源量に基づく）。すなわち、DBMS 4 1 2 は、新たに生成するタスクに、結果生成に至るまでの後続の DB オペレーションを対応させ、実行する。これにより、一つのタスクにより、そのタスクに対応した結果生成に至るまでの後続の DB オペレーションを、途中でメモリ資源の割り当てを行うことなく実行することができる。なお、結果生成とは、クエリ実行プランに基づき定義された処理であり、典型的には、クエリの発行元に返される結果の生成である。だがそれに限らず、例えば、クエリ実行プランに含まれる部分クエリ毎の中間結果の生成でも良い。結果生成に至るまでの後続の DB オペレーションとは、新たに生成するタスクにより実行を開始する DB オペレーションから、該 DB オペレーションより後の結果生成に至るまでの一連の DB オペレーションを意味する。

また、実施例 2 では、DBMS 4 1 2 は、タスクを新たに生成する都度に、新たに生成するタスクを実行するために必要な情報を有するコンテキストを生成する。コンテキストは、例えば、新たに生成するタスクにおいて実行を開始する DB オペレーションが、クエリ実行プランに含まれる情報である 1 以上の DB オペレーションのうちのいずれであるかに関する第 1 の情報と、該第 1 の情報が示す DB オペレーションに要するデータのアクセス先に関する第 2 の情報と、結果を生成するために必要なフェッチ済みのデータに関する第 3 の情報とを含む情報である。DBMS 4 1 2 は、生成したコンテキストを基に新たなタスクを生成して実行する。実施例 2 では、コンテキストを生成することにより、タスクの生成を保留するタスクは生じない。

【 0 1 2 9 】

図 2 6 は、実施例 2 において、DBMS 4 1 2 が図 3 のクエリ 1 を図 5 のクエリ実行プランに従って実行する際に、メモリ資源の枯渇を回避できることを説明するための模式図の一例である。表記のルールは、基本的に図 7 と同様であるが、更に次のルールに従う。
 (*) 図上部の横に長い五角形は、一つのタスクに対応した、結果生成に至るまでの後続の DB オペレーションを意味する。

(*) 五角形内部の文字列について、単位時間内（例えば、 $t_0 \sim t_1$ 、又は、 $t_1 \sim t_2$ 等）の「|」で区切られる各フィールドは、該当タスクに対応した、結果生成に至るまでの後続の DB オペレーションに要するメモリ資源に相当する。メモリ資源に相当する各フィールドには、該タスクを実行する時間の経過に伴って、単位時間経過後にタスク実行によりフェッチされるデータ、又は、結果を生成するために必要なフェッチ済みのデータが入力される。すなわち、一つのタスクに対応した、結果生成に至るまでの後続の DB オペレーションに要するメモリ資源は、タスク生成時に割り当てられることを表す。そして、そのメモリ資源に、時間の経過とともにデータが入力されることを表す。

(*) 五角形内部の文字列について、太字及び下線の文字列は、該当タスクの単位時間に対応する DB オペレーションにおいて必要なデータである。一方、細字の文字列は、該当タスクの単位時間より前に対応した DB オペレーションで得たデータであり、その単位時間以降において必要のないデータである。故に、該データに関するメモリ資源は、その単

10

20

30

40

50

位時間以降において解放可能であることを表す。

(*) 小さい太枠の正方形は、コンテキストを表す。コンテキストを基にタスクを生成することを、タスクの実行開始時点を表す五角形の左端にコンテキストを配置することによって表す。

(*) 一つのタスクの実行に要するメモリ資源量は、簡略化のため、タスクを表す五角形の内部のフィールドの数がいくつであっても、図7と同じく「1」とする。

(*) コンテキストに要するメモリ資源量は、タスクの実行に要するメモリ資源量に比べて十分小さいので本説明では「0」とする。

【0130】

図26によれば、実施例2においても、DBMS412は、割り当て上限値(割り当て可能総メモリ資源量909)を超える量のメモリ資源の割り当てを回避することができる。例えば、DBMS412は、図5のクエリ実行プランに従って下記の通り実行する。なお、下記の同時生成タスク数決定処理における第1のメモリ資源量(新たに生成するタスク当たり割り当てを要するメモリ資源量)は、上記の表記ルールの通り「1」である。

(t0) 索引Aにアクセスする新たなタスクを実行するための一つのコンテキスト25Aを生成する。DBMS412は、コンテキスト25Aに基づき同時生成タスク数決定処理を行う。例えば、対象のコンテキストの数と同数である生成可能タスク数「1」と、第1のメモリ資源量「1」と、第2のメモリ資源量「6」(割り当て上限値「6」)とに基づき、同時生成タスク数を「1」と算出する。DBMS412は、算出した同時生成タスク数と同数のタスク24Aを生成する。そして、コンテキスト25Aに基づいて、対応するDBオペレーションに要するメモリ資源をタスク24Aに割り当てて実行する。タスク24Aに割り当てられるメモリ資源は、例えば、図5のクエリ実行プランにおいて、索引Aへのアクセスから結果生成に至るまでの全てのDBオペレーションに要するデータに相当し、五つのフィールドに相当する。

(t1) タスク24Aの実行結果から、DBMS412は、表Aの二つのレコードを含むデータそれぞれをフェッチする必要があることが分かる。そこで、DBMS412は、表Aの一方のレコード(3レコード目)を含むデータをフェッチするためのタスクについて、該タスクを実行するためのコンテキスト25Bを一つ生成する。また、DBMS412は、表Aの他方のレコード(1レコード目)を含むデータのフェッチをタスク24Aにより継続して行う。タスク24Aの継続実行は、対応するDBオペレーションに必要なメモリ資源を予め割り当てているため可能である。DBMS412は、コンテキスト25Bに基づき同時生成タスク数決定処理を行う。例えば、対象のコンテキストの数と同数である生成可能タスク数「1」と、第1のメモリ資源量「1」と、第2のメモリ資源量「5」(割り当て上限値「6」から割り当て済みのメモリ資源量「1」を減算した値)とに基づき、同時生成タスク数を「1」と算出する。DBMS412は、算出した同時生成タスク数と同数のタスク24Bを生成する。そして、コンテキスト25Bに基づいて、対応するDBオペレーションに要するメモリ資源をタスク24Bに割り当てて実行する。タスク24Bに割り当てられるメモリ資源は、例えば、図5のクエリ実行プランにおいて、表Aのデータフェッチから結果生成に至るまでの一連のDBオペレーションに要するデータに相当し、四つのフィールドに相当する。

(t2) タスク24Aの実行結果から、DBMS412は、索引BにアクセスしてRowID Listを一つ検索する必要があることが分かる。DBMS412は、該当するDBオペレーションをタスク24Aにより継続して行う。同様に、タスク24Bの実行結果から、DBMS412は、索引BにアクセスしてRowID Listを一つ検索するDBオペレーションをタスク24Bにより継続して行う。

(t3) タスク24Aの実行結果から、DBMS412は、表Bの三つのレコードを含むデータそれぞれをフェッチする必要があることが分かる。そこで、DBMS412は、表Bの二つのレコードを含むデータそれぞれをフェッチするためのタスク各々について、それらタスクを実行するためのコンテキストを二つ生成する。また、DBMS412は、表Bの残り一つのレコードを含むデータのフェッチをタスク24Aにより継続して行う。同

10

20

30

40

50

様に、タスク 24B の実行結果から、DBMS 412 は、表 B の五つのレコードを含むデータそれぞれをフェッチする必要があることが分かる。そこで、DBMS 412 は、表 B の四つのレコードを含むデータそれぞれをフェッチするためのタスク各々について、それらタスクを実行するためのコンテキストを四つ生成する。また、DBMS 412 は、表 B の残り一つのレコードを含むデータのフェッチをタスク 24B により継続して行う。DBMS 412 は、生成した計六つのコンテキストに基づき同時生成タスク数決定処理を行う。例えば、対象のコンテキストの数と同数である生成可能タスク数「6」と、第1のメモリ資源量「1」と、第2のメモリ資源量「4」（割り当て上限値「6」から割り当て済みのメモリ資源量「2」を減算した値）とに基づき、同時生成タスク数を「4」と算出する。DBMS 412 は、算出した同時生成タスク数と同数のタスク 24C、24D、24E 及び 24F を生成する。そして、生成した六つのコンテキストのうち四つに基づいて、対応する DB オペレーションに要するメモリ資源をタスク 24C、24D、24E 及び 24F 各々に割り当てて実行する。タスク 24C、24D、24E 及び 24F 各々に割り当てられるメモリ資源は、例えば、図 5 のクエリ 1 のクエリ実行プランにおいて、表 B のデータフェッチから結果生成に至るまでの一連の DB オペレーションに要するデータに相当し、二つのフィールドに相当する。

10

(t4) t3 で実行していた六つのタスクについて、その実行を終え、クエリの結果を生成する。DBMS 412 は、実行を終えた六つのタスクそれぞれについて、割り当てていたメモリ資源を解放しタスクを終了する(t4の直後)。これにより、第2のメモリ資源量が「6」となるため、DBMS 412 は、タスク生成に用いていない残り二つのコンテキストに基づき同時生成タスク数決定処理を行う。DBMS 412 は、該処理により同時生成タスク数を「2」と算出し、算出した数と同数のタスクを生成する。そして、生成した二つのタスクについて、同時生成タスク数決定処理に用いた二つのコンテキストに基づき、対応する DB オペレーションに要するメモリ資源を割り当てて実行する。

20

(t5) t4 までに実行した全てのタスクの実行を終え、クエリの結果を生成する。

【0131】

このように、実施例 2 においても、実施例 1 と同様に、生成された全てのタスクを並行に実行しても、割り当てられるメモリ資源の量は、割り当て上限値「6」以下となる。すなわち、タスクに割り当てられるメモリ資源が枯渇してしまうことを回避することができる。上記の通り、同時に生成されるタスクの数を同時生成タスク数と同じにすることで、割り当て上限値の範囲内でデータ読出し要求を最も高い多重度を以って発行することが期待でき、よって、クエリ実行時間の短縮化が期待できる。

30

更に、実施例 2 では、タスク 24A 及び 24B のように、タスクの実行を開始したら結果生成に至るまで、途中でメモリ資源の割り当てを行うことなく実行することができる。メモリ資源の割り当て処理では、図 16 の通りタスクの実行待ちが生じる可能性があるため、その処理をタスクの実行中に行わないことで、開始したタスクを効率的に実行することができる。また、新たに生成するタスクの実行基となるコンテキストを生成しておくことで、割り当て済みのメモリ資源を有したままとなる、タスクの生成を待つタスクは生じない。これにより、タスクによる非効率なメモリ資源の浪費が起こることなくクエリを実行することができる。なお、DBMS 412 は、一つのタスクの実行結果から生成するコンテキスト(例えば、t3 でタスク 24A の実行結果から生成する二つのコンテキスト)を、まとめて一つのコンテキストとして管理しても良い。

40

【0132】

また、実施例 2 においても、実施例 1 と同様に、DBMS 412 が複数のクエリを受け付け、受け付けた複数のクエリを並行して実行する場合にも、タスクに割り当てられるためのメモリ資源の枯渇を回避することができる。図 27 は、DBMS 412 が図 3 のクエリ 1 と図 4 のクエリ 2 を同時に受け付け、その二つのクエリを図 5 及び図 6 のクエリ実行プランに従って並行に実行する際の模式図の例を示す。表記のルールは図 26 と同様である。実施例 1 と同様、DBMS 412 は、実行するクエリ(部分クエリ)毎に優先度を設ける。図 27 に示すように、DBMS 412 は、それら優先度を基に、DBMS 412 は、割

50

り当て上限値「6」を、実行する部分クエリ毎に配分する。そして、各部分クエリの実行において、タスクを新たに生成する都度に同時生成タスク数決定処理を行う。すなわち、DBMS 412は、部分クエリ毎に、その実行に配分された割り当て可能なメモリ資源量の上限值を基に、同時生成タスク数を決定する。各部分クエリの実行において、新たに生成するタスクの数を算出した同時生成タスク数以下とすることで、タスクに割り当てるためのメモリ資源の枯渇を回避することができる。

また、図26及び図27は、図7～図9と同様に、実施例2における概要のイメージを表す模式的な図であり、DBMS 412は、複数のタスクを同じ時点で開始しなくても良い。

【0133】

以下、実施例2を詳細に説明する。

【0134】

図28は、実施例2に係るクエリ実行部423の構成を示す。

【0135】

クエリ実行部423は、更に、コンテキスト管理部1001を有する。コンテキスト管理部1001は、タスクの実行結果に基づき生成されるコンテキストを管理する。

【0136】

図29は、実施例2にて、DBMS 412が受け付けたクエリの実行に際して、タスクを生成し実行する処理の全体の流れを示す。

【0137】

S2701及びS2702で、クエリ実行部423は、図12のS1001及びS1002と同じ処理を行う。S2702で、クエリ実行部423は、例えば、複数のタスクにより並行して実行可能な部分クエリの数を生可能タスク数として算出する。

【0138】

S2703で、クエリ実行部423(コンテキスト管理部1001)は、S2702で算出した生成可能タスク数と同数のコンテキストを生成する。生成された各コンテキストは、部分クエリ毎の先頭のDBオペレーションを行うタスクにそれぞれ対応し、該タスクを実行するためのものである。コンテキストの具体的な生成処理では、クエリ実行部423(メモリ資源管理部425)は、コンテキストに要するメモリ資源の量を算出し、算出した量のメモリ資源を予約する。そして、該当する部分クエリの実行用途に割り当てる(図18のメモリ資源予約処理及び図16のメモリ資源割り当て処理に相当する処理を行う)。クエリ実行部423は、コンテキスト(割り当てたメモリ資源)に、新たに生成するタスクの実行に必要な情報を設定し、コンテキスト管理部1001で管理する。図18及び図16に相当する処理を経ることにより、コンテキストとして使用するメモリ資源の量は、クエリ実行管理表424内の対応する部分クエリの既予約メモリ資源量907及び既割り当てメモリ資源量906として管理される。なお、コンテキストに要するメモリ資源の量は、新たに生成するタスク当たり割り当てを要するメモリ資源量である第1のメモリ資源量に比べて十分小さい。そのため、コンテキストに要するメモリ資源の予約及び割り当ては、多くのケースで成功する。万一コンテキストに要するメモリ資源の予約又は割り当てに失敗した場合、クエリ実行部423(メモリ資源管理部425)は、そのメモリ資源量に達するまで予約又は割り当て処理を繰り返す。その際、或る部分クエリの実行開始時間904から所定の時間を経過した場合、クエリ実行部423は、該部分クエリを含むクエリの実行をエラー終了しても良い。

【0139】

S2704で、クエリ実行部423(同時生成タスク数決定部431)は、S2703で生成したコンテキストを基に、同時生成タスク数決定処理を行う。なお、実施例2に係る同時生成タスク数決定処理では、第1のメモリ資源量を、S2703で生成したコンテキストを基に新たに生成するタスクに対応した、結果生成に至るまでの後続のDBオペレーションに要するメモリ資源の量に基づき算出する。この第1のメモリ資源量の算出以外において、S2704で行われる同時生成タスク数決定処理は、実施例1で述べた図15

10

20

30

40

50

の同時生成タスク数決定処理と実質的に同じである。

【0140】

S2705で、クエリ実行部423は、図12のS1004と同じ処理を行い、S2704で決定された同時生成タスク数と同数のタスクを生成する。

【0141】

S2706で、クエリ実行部423は、S2705で生成した各々のタスクについて、S2703で生成した各コンテキストに基づきタスク実行処理を並行して行う。実施例2のタスク実行処理は、コンテキストを生成し用いる点で実施例1のタスク実行処理と異なる。その詳細は図30に示す。なお、コンテキストに基づいてタスクの実行を開始する時点で、クエリ実行部423は、該コンテキストのメモリ資源を解放して良い(図17のメモリ資源解放処理を行う)。クエリ実行部423は、上記各々のタスクについてタスク実行処理を開始した後、図29の処理を終える。

10

【0142】

図30は、実施例2に係るタスク実行処理(図29のS2706、図31のS2909、図32のS3005)の流れを示す。図30では、一つのタスク(図30の説明において「対象タスク」と言う)を例に取った流れを示す。

【0143】

S2801で、クエリ実行部423(メモリ資源管理部425)は、メモリ資源割り当て処理を行う。本処理は、対象タスクを実行するためのコンテキスト(図29のS2703、又は、図31のS2906で生成されたコンテキスト)を基に行われる。クエリ実行部423(メモリ資源管理部425)は、図16に示したメモリ資源割り当て処理と実質的に同じ処理を行う。本処理において、新規割り当てメモリ資源量(対象タスクを実行するために必要とするメモリ資源量)は、コンテキストを基に特定されるメモリ資源量であって、対象タスクに対応した、結果生成に至るまでの後続のDBオペレーションを実際に実行するためのメモリ資源量である。

20

【0144】

S2802で、クエリ実行部423は、対象タスクを実行するためのコンテキストを基に、該タスクに対応するDBオペレーションを実行する処理を行う。本処理をDBオペレーション処理と言う。DBオペレーション処理の詳細は、図31に示す。

【0145】

S2803で、クエリ実行部423(メモリ資源管理部425)は、DBオペレーション処理を終えた対象タスクについて、解放可能なメモリ資源を解放する処理を行う。本処理は、図17に示したメモリ資源解放処理と同じである。

30

【0146】

S2804で、クエリ実行部423は、S2803での処理による未予約メモリ資源量908の増加に伴い、タスクを追加生成するタスク追加生成処理を行う。本処理は、コンテキスト管理部1001により管理されている、タスクの実行に用いられていないコンテキストを基に行われる。タスク追加生成処理の詳細は、図32に示す。クエリ実行部423は、S2804の処理を開始するとともに、対象タスクの終了を実行タスク管理部426に指示して良い。

40

【0147】

図31は、実施例2に係るDBオペレーション処理(図30のS2802、図31のS2911)の流れを示す。図31の説明において、処理する対象のDBオペレーションを「処理対象DBオペレーション」と言い、処理対象DBオペレーションを含む部分クエリを「対象部分クエリ」と言う。また、処理対象DBオペレーションに対応するタスクを「対象タスク」と言う。

【0148】

S2901で、クエリ実行部423は、図13のS1102と同様の処理を行い、処理対象DBオペレーションを実行する。DBオペレーションを実行するための情報は、実施例1では、タスクの生成元となったタスクが有する情報に基づくが、実施例2では、コン

50

テキスト及び対象タスクが有する情報に基づく。

【0149】

S 2 9 0 2 で、クエリ実行部 4 2 3 は、図 1 3 の S 1 1 0 3 と同様の処理を行い、S 2 9 0 1 での処理対象 DB オペレーションの実行結果に基づき、生成可能タスク数を算出する。すなわち、クエリ実行部 4 2 3 は、クエリ実行プランにおいて、処理対象 DB オペレーションの次の DB オペレーションを行うために、新たに生成可能なタスク数を算出する。

【0150】

S 2 9 0 3 で、クエリ実行部 4 2 3 は、S 2 9 0 2 で算出された生成可能タスク数が 1 以上であるか否かを判断する。クエリ実行部 4 2 3 は、該判断の結果が肯定的であれば、S 2 9 0 4 を行い、該判断の結果が否定的であれば、処理を終える。

10

【0151】

S 2 9 0 4 で、クエリ実行部 4 2 3 は、コンテキストを生成可能か否か判断する。該判断は、例えば、下記 (x) 及び (y) の基準、

(x) 対象部分クエリの実行に関するコンテキストの総数の、所定数に対する比率、

(y) コンテキストに要するメモリ資源の予約可否 (図 1 8 のメモリ資源予約処理に従う) 、

を基に行われる。例えば、(x) の比率が所定比率を超えている、(y) の予約が不可能である、のいずれかに該当すれば、クエリ実行部 4 2 3 は、S 2 9 0 4 の判断の結果を否定的とし、上記のいずれにも該当しなければ、S 2 9 0 4 の判断の結果を肯定的として良い。クエリ実行部 4 2 3 は、該判断の結果が肯定的であれば、S 2 9 0 5 を行い、該判断の結果が否定的であれば、S 2 9 1 2 を行う。

20

【0152】

S 2 9 0 5 で、クエリ実行部 4 2 3 は、対象部分クエリにおいて、処理対象 DB オペレーションより後続の結果生成に至るまでの DB オペレーションの数と、該部分クエリの先頭の DB オペレーションから結果生成に至るまでの DB オペレーションの数との比率が、所定値より大きいか否かを判断する。該比率が大きい場合、結果生成に至るまでに多くの DB オペレーションが存在する。一方、該比率が小さい場合、結果生成に至るまでの DB オペレーションの数は少ない。従って、該比率が大きい場合、DB オペレーションの数に応じてタスクを動的に生成できる可能性が向上し、データ読出し要求の多重化 (クエリ実行時間の短縮化) を期待できる。一方、該比率が小さい場合、タスクを動的に生成できる可能性は該比率が大きい場合と比べて小さくなる。クエリ実行部 4 2 3 は、S 2 9 0 5 の判断の結果が肯定的であれば、S 2 9 0 6 を行い、S 2 9 0 5 の結果が否定的であれば、S 2 9 1 2 を行う。

30

【0153】

S 2 9 0 6 で、クエリ実行部 4 2 3 は、S 2 9 0 2 で算出した生成可能タスク数より 1 少ない数のコンテキストを生成する。生成された各コンテキストは、S 2 9 0 1 の実行結果に基づいて、処理対象 DB オペレーションの次の DB オペレーションを行うタスクにそれぞれ対応し、該タスクを実行するための情報である。コンテキストの生成処理は、図 2 9 の S 2 7 0 3 の処理と同様である。クエリ実行部 4 2 3 は、上記した次の DB オペレーションの残した一つ分について、対象タスクにより継続して実行する (後述の S 2 9 1 1) 。なお、生成するコンテキストの数がゼロの場合 (生成可能タスク数が 1 である場合) 、クエリ実行部 4 2 3 は、以下の S 2 9 0 7 、S 2 9 0 8 、及び、S 2 9 0 9 のステップでは何も行わず、S 2 9 1 0 のステップに進む。

40

【0154】

S 2 9 0 7 で、クエリ実行部 4 2 3 は、S 2 9 0 6 で生成したコンテキストを基に、図 2 9 の S 2 7 0 4 と同様の、同時生成タスク数決定処理を実行する。

【0155】

S 2 9 0 8 で、クエリ実行部 4 2 3 は、図 1 2 の S 1 0 0 4 と同じ処理を行い、S 2 9 0 7 で決定された同時生成タスク数と同数のタスクを生成する。

50

【 0 1 5 6 】

S 2 9 0 9 で、クエリ実行部 4 2 3 は、S 2 9 0 8 で生成された各タスクについて、S 2 9 0 6 で生成した各コンテキストに基づきタスク実行処理（図 3 0）を並行に実行する。

【 0 1 5 7 】

S 2 9 1 0 で、クエリ実行部 4 2 3 は、対象タスクにより次の S 2 9 1 1 のステップを行う上で、該タスクに割り当てられたメモリ資源のうち解放可能なメモリ資源を解放する（図 1 7 のメモリ資源解放処理を行う）。例えば、図 2 6 に示した、タスクによる実行を表す五角形内部の文字列について、ある単位時間に文字列が細字であるフィールドに相当するメモリ資源は、解放されても良い。なぜなら、該メモリ資源に格納されたデータは、当該単位時間より前に実行された D B オペレーションの結果であり、結果の生成に用いられないデータであるからである。

10

【 0 1 5 8 】

S 2 9 1 1 で、クエリ実行部 4 2 3 は、S 2 9 0 6 で一つ残しておいた、処理対象 D B オペレーションの次の D B オペレーションを、対象タスクに対応させて継続して実行する。すなわち、クエリ実行部 4 2 3 は、本図 3 1 に示す D B オペレーション処理を再び行う。

【 0 1 5 9 】

S 2 9 0 4 及び S 2 9 0 5 の判断の結果が否定的であるとき、S 2 9 1 2 で、クエリ実行部 4 2 3 は、処理対象 D B オペレーションの実行結果に関連する全ての D B オペレーションを、対象タスク一つに対応させて実行する。このとき、クエリ実行部 4 2 3 は、新たにコンテキスト及びタスクを生成しない。対象タスクには、処理対象 D B オペレーションより後続の結果生成に至るまでの D B オペレーションについて、それら D B オペレーションに必要なメモリ資源を割り当てている。そのため、対象タスクにより、処理対象 D B オペレーションの実行結果に関連する全ての D B オペレーションの実行が可能となる。なお、S 2 9 1 2 において、クエリ実行部 4 2 3 は、上記した関連する全ての D B オペレーションにおける各実行結果に基づいて、S 2 9 0 4 及び S 2 9 0 5 の判断を行っても良い。このとき、クエリ実行部 4 2 3 は、或る D B オペレーションの実行結果から、S 2 9 0 4 及び S 2 9 0 5 の判断を踏まえてコンテキストの生成が可能になった場合、S 2 9 0 6 ~ S 2 9 1 1 に従う処理を行っても良い。すなわち、クエリ実行部 4 2 3 は、再びコンテキストを生成し、該コンテキストに基づいてタスクを生成し実行しても良い。

20

30

【 0 1 6 0 】

図 3 2 は、実施例 2 に係るタスク追加生成処理（図 3 0 の S 2 8 0 4）の流れを示す。

【 0 1 6 1 】

タスク追加生成処理は、実施例 1 の図 2 5 と同様に、例えば、下記（ 1 ）～（ 3 ）の事象のうち少なくとも一つが発生した契機に開始される処理である。

（ 1 ）クエリ実行管理表 4 2 4 の割り当て可能総メモリ資源量 9 0 9 又は未予約メモリ資源量 9 0 8 が増加した。

（ 2 ）クエリ実行管理表 4 2 4 のある部分クエリの優先度 9 0 1 が変化した、又は、最優先フラグ 9 0 2 がオンになった。

40

（ 3 ）一定時間が経過した。

なお、本タスク追加生成処理は、クエリ実行管理表 4 2 4 の未予約メモリ資源量 9 0 8 が所定値以上の場合に、クエリ実行管理表 4 2 4 の全ての部分クエリについて、タスクの実行に用いられていないコンテキストが存在しなくなるまで繰り返されても良い。

【 0 1 6 2 】

S 3 0 0 1 で、クエリ実行部 4 2 3 は、クエリ実行管理表 4 2 4 を参照し、タスクの実行に用いられていないコンテキストが存在する部分クエリのうち、優先度 9 0 1 が最も高い部分クエリを選択する。

【 0 1 6 3 】

S 3 0 0 2 で、クエリ実行部 4 2 3 は、S 3 0 0 1 で選択した部分クエリの実行におい

50

て生成されたコンテキスト群のうち、実行優先度の高いコンテキストを1以上選択する。例えば、クエリ実行部423は、コンテキストに対応するタスクで実行するDBオペレーションについて、該DBオペレーションより後続のDBオペレーションの数が最も少ないコンテキストを1以上選択する。

【0164】

S3003で、クエリ実行部423は、S3002で選択した1以上のコンテキストを基に、図29のS2704と同様の、同時生成タスク数決定処理を行う。

【0165】

S3004で、クエリ実行部423は、図12のS1004と同じ処理を行い、S3003で決定された同時生成タスク数と同数のタスクを生成する。

10

【0166】

S3005で、クエリ実行部423は、S3004で生成された各々のタスクについて、S3002で選択した各コンテキストを基に、図30に示したタスク実行処理を行う。なお、コンテキストに基づいてタスクの実行を開始する時点で、クエリ実行部423は、該コンテキストのメモリ資源を解放して良い(図17のメモリ資源解放処理を行う)。

【実施例3】

【0167】

以下、実施例3を説明する。その際、実施例1及び2との相違点を主に説明し、実施例1及び2との共通点については説明を省略或いは簡略する。

【0168】

20

図33は、実施例3に係る計算機システムの構成を示す。

【0169】

アプリケーションサーバ(以下、APサーバ)3102は、DBサーバ401に、通信ネットワーク3112を介して通信できるように接続されている。また、DBサーバ401は、外部ストレージ装置402に、通信ネットワーク403を介して通信できるように接続されている。ユーザ端末(クライアント端末)3101は、APサーバ3102に、通信ネットワーク3111を介して通信できるように接続されている。DBサーバ401は、DB451を管理するDBMS412を実行する。外部ストレージ装置402は、DB451を格納する。APサーバ3102は、DBサーバ401で実行されるDBMS412に対してクエリを発行するAPを実行する。ユーザ端末3101は、APサーバ3102で実行されるAPに要求を出す。なお、ユーザ端末3101、又は、APサーバ3102は、複数存在しても良い。

30

【0170】

APサーバ管理端末3103は、通信ネットワーク3114を介してAPサーバ3102に接続されている。DBサーバ管理端末3104は、通信ネットワーク3115を介してDBサーバ401に接続されている。ストレージ管理端末3105は、通信ネットワーク3116を介して外部ストレージ装置402に接続されている。APサーバ管理端末3103は、APサーバ3102を管理する端末である。DBサーバ管理端末3104は、DBサーバ401を管理する端末である。ストレージ管理端末3105は、外部ストレージ装置402を管理する端末である。DBサーバ管理者又はユーザは、DBサーバ管理端末3104から、クエリ実行管理表424内の部分クエリに対応する優先度901及び最優先フラグ902等を設定して良い。なお、管理端末3103~3105のうちの少なくとも二つが共通(一体)であっても良い。また、通信ネットワーク3111、3112、3114、3115、3116、及び403のうちの少なくとも二つが共通(一体)であっても良い。

40

【0171】

実施例3では、例えば、下記の通り処理が実行される。

(S31) ユーザ端末3101は、APサーバ3102に要求(以下、ユーザ要求)を発行する。

(S32) APサーバ3102は、S31で受信したユーザ要求に従いクエリを生成する

50

。そして、生成したクエリをDBサーバ401に発行する。

(S33) DBサーバ401は、APサーバ3102からのクエリを受け付け、受け付けたクエリを実行する。DBサーバ401は、受け付けたクエリの実行において必要なデータ入出力要求(例えばデータ読み出し要求)を外部ストレージ装置402に発行する。DBサーバ401は、一つのクエリの実行において、複数のデータ入出力要求を並行して発行することがある。そのため、DBサーバ401は、一つのクエリの実行において、S33の要求を複数回並行して行うことがある。

(S34) 外部ストレージ装置402は、S33で発行されたデータ入出力要求について、DBサーバ401に回答する。外部ストレージ装置402は、S34の回答を複数回並行して行うことがある。

(S35) DBサーバ401は、クエリの実行結果を生成し、APサーバ3102に送信する。

(S36) APサーバ3102は、クエリの実行結果を受信する。そして、該実行結果に従う、S31で受信したユーザ要求に対する回答を、ユーザ端末3101に送信する。

なお、APサーバ3102に発行されるユーザ要求、又は、DBサーバへ発行されるクエリは、同時に複数あっても良い。

【0172】

以上、幾つかの実施例を説明したが、本発明は、これらの実施例に限定されるものでなく、その要旨を逸脱しない範囲で種々変更可能であることはいうまでもない。

【符号の説明】

【0173】

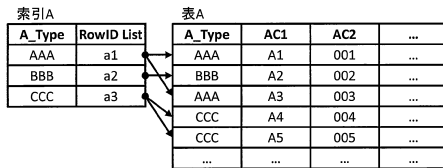
412...データベース管理システム(DBMS)

10

20

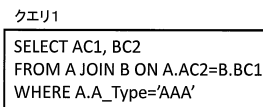
【図1】

Fig. 1



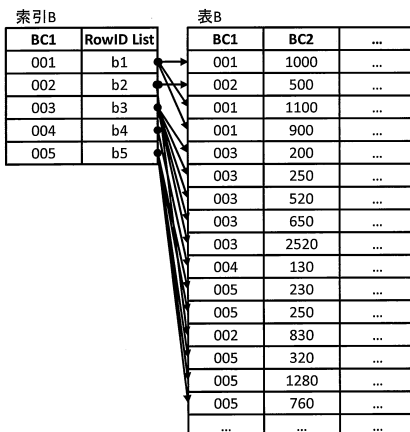
【図3】

Fig. 3



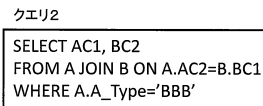
【図2】

Fig. 2

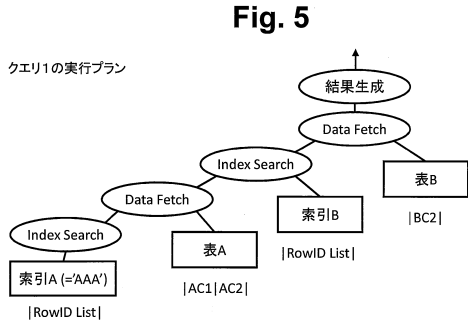


【図4】

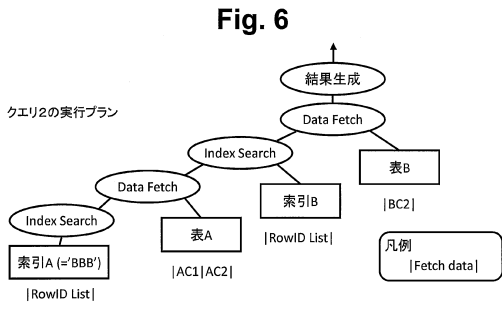
Fig. 4



【 図 5 】



【 図 6 】



【 図 7 】

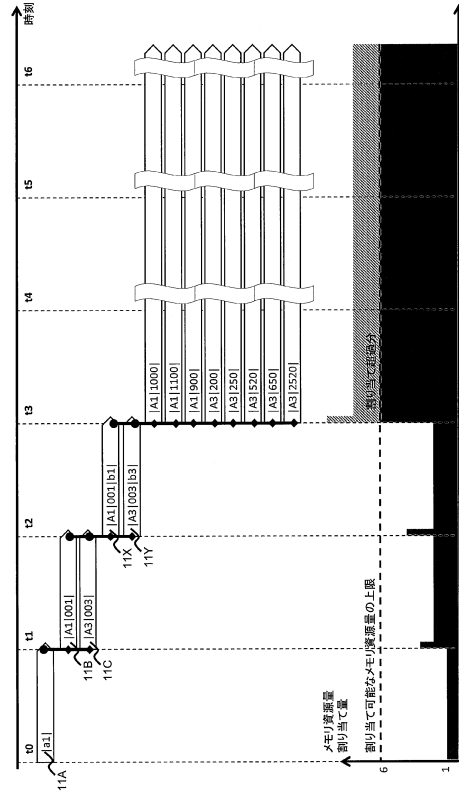
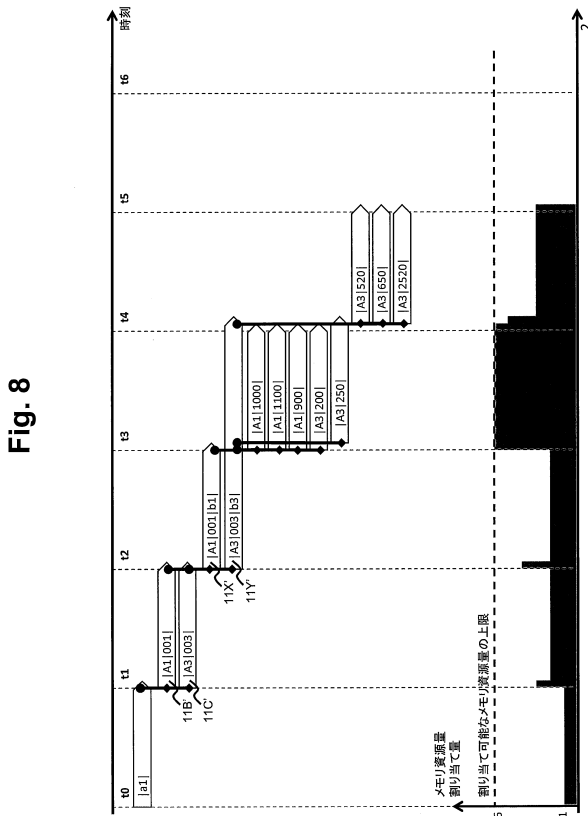
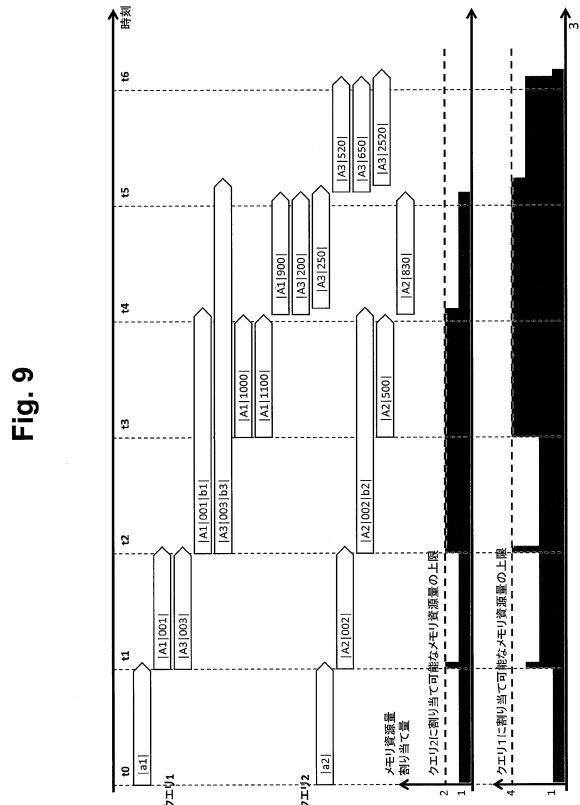


Fig. 7

【 図 8 】

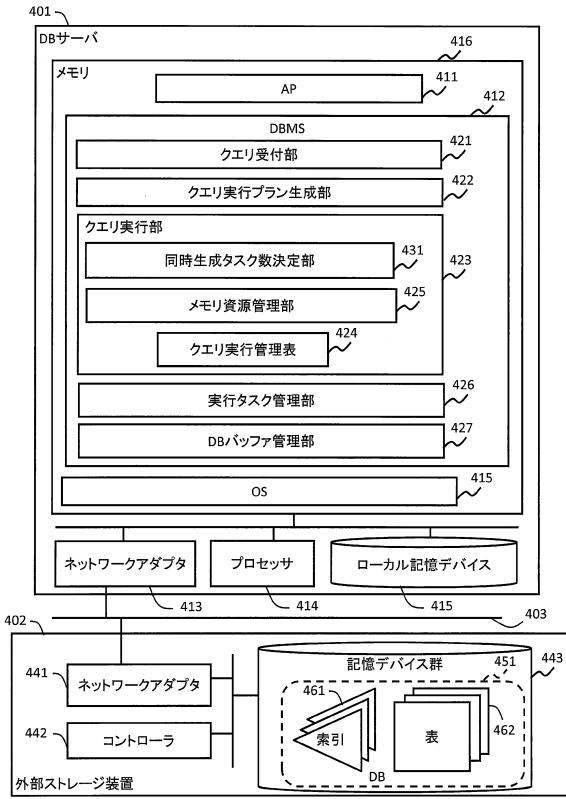


【 図 9 】



【図10】

Fig. 10



【図11】

Fig. 11

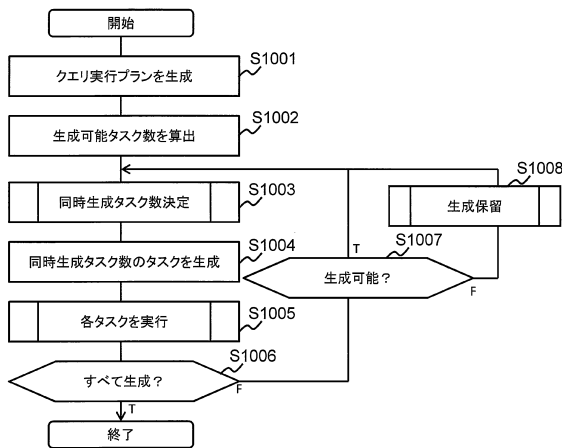
クエリ実行管理表
424

実行中の部分クエリ	優先度	最優先フラグ	目標実行時間	実行開始時刻	実行達成率
部分クエリ1	3	-	60 min.	14:32:11	25%
部分クエリ2	2	-	30 min.	14:15:39	30%
部分クエリ3	5	-	10 min.	14:35:41	12%

既割り当てメモリ資源量	既予約メモリ資源量	未予約メモリ資源量	割り当て可能総メモリ資源量
30MB	30MB	5MB	100MB
10MB	20MB		
40MB	45MB		

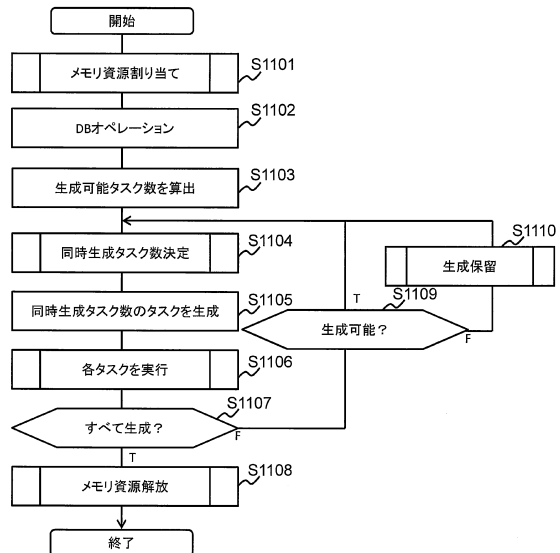
【図12】

Fig. 12



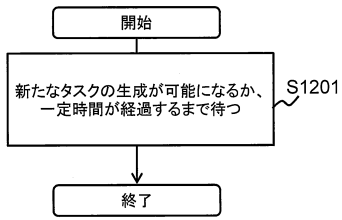
【図13】

Fig. 13



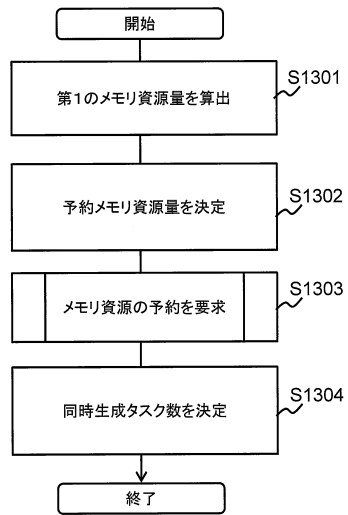
【図 14】

Fig. 14



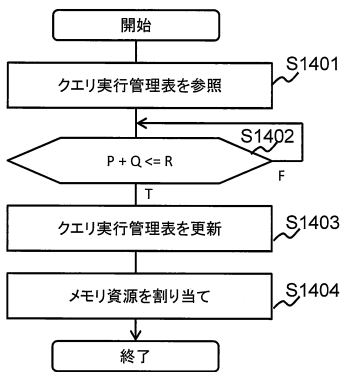
【図 15】

Fig. 15



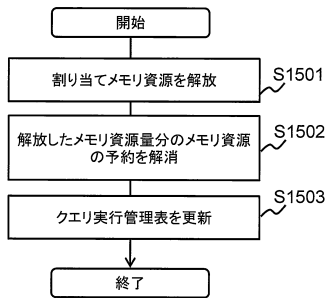
【図 16】

Fig. 16



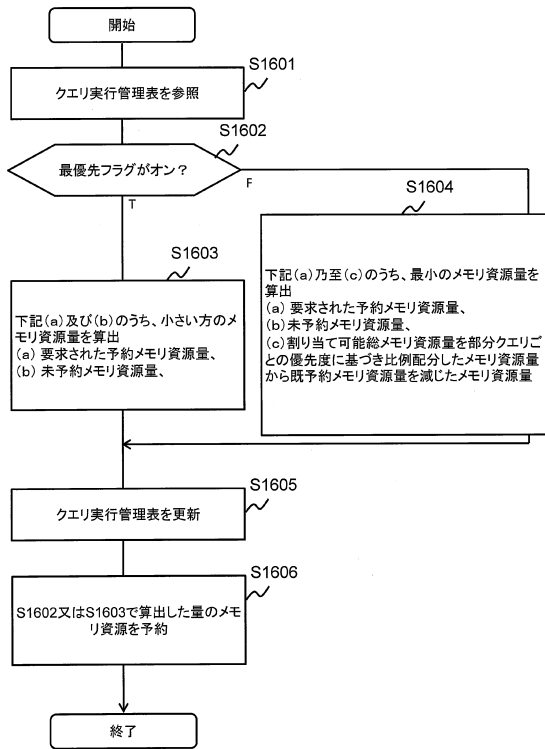
【図 17】

Fig. 17



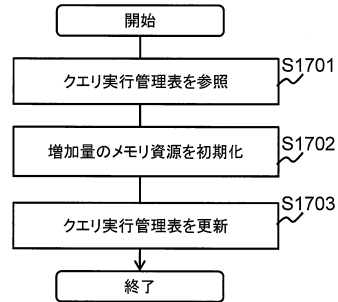
【図18】

Fig. 18



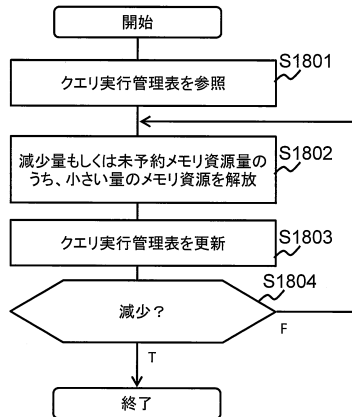
【図19】

Fig. 19



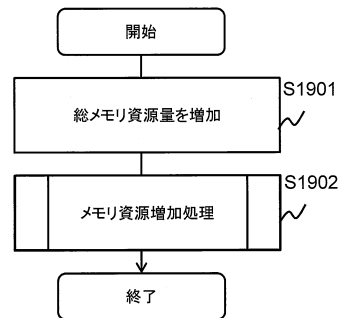
【図20】

Fig. 20

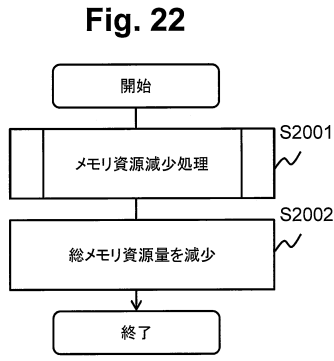


【図21】

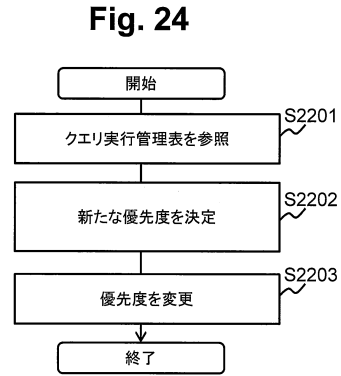
Fig. 21



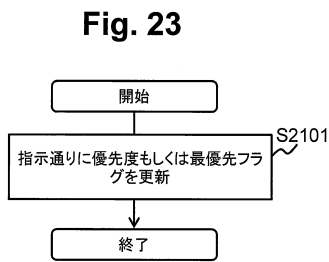
【図22】



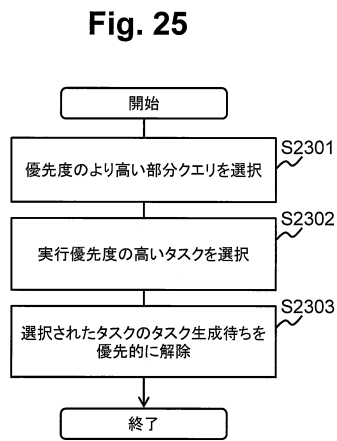
【図24】



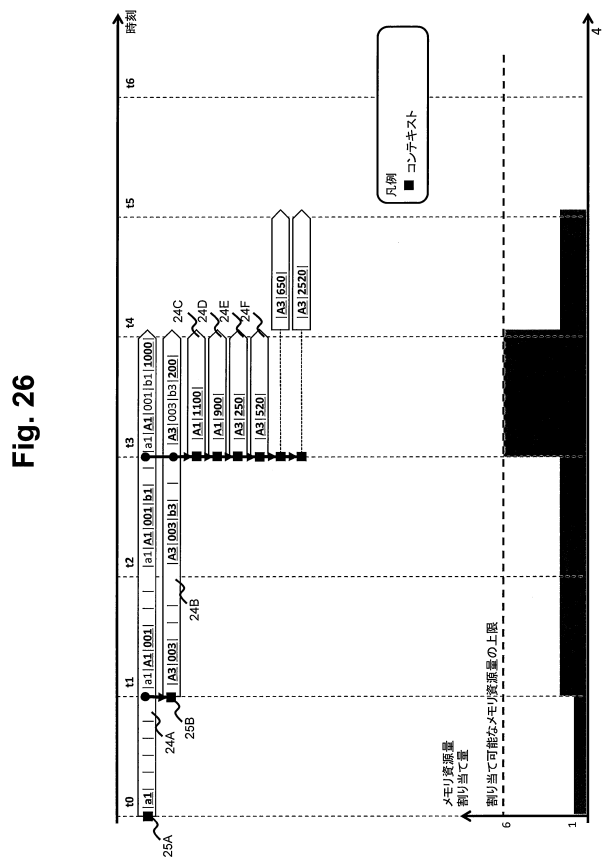
【図23】



【図25】

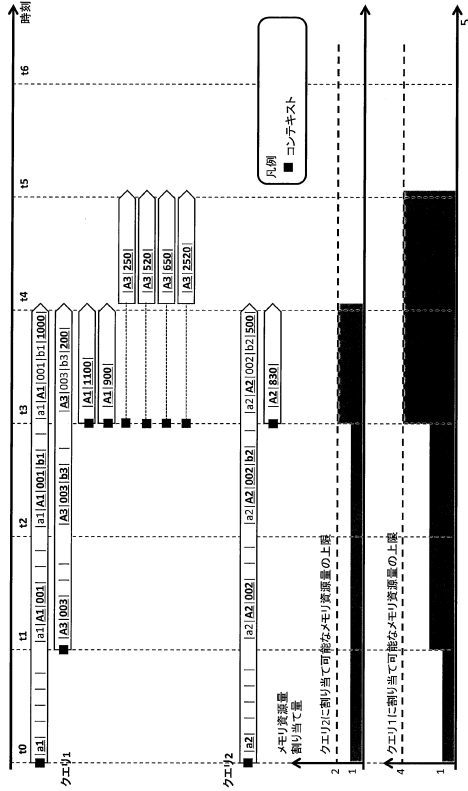


【図26】



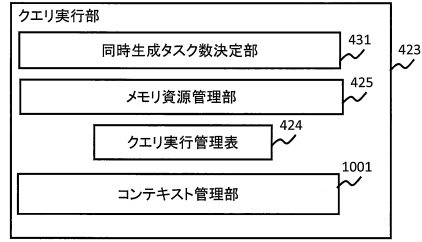
【 図 27 】

Fig. 27



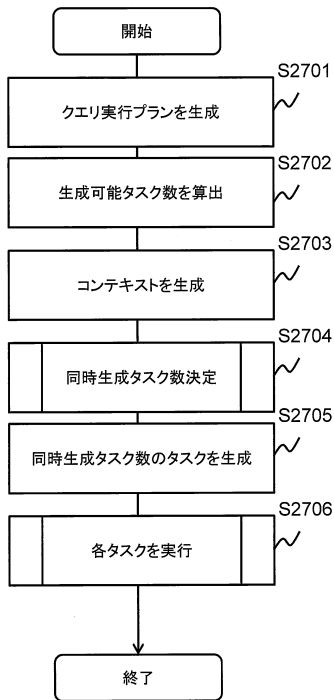
【 図 28 】

Fig. 28



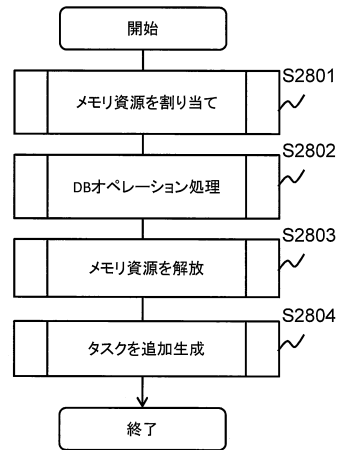
【 図 29 】

Fig. 29

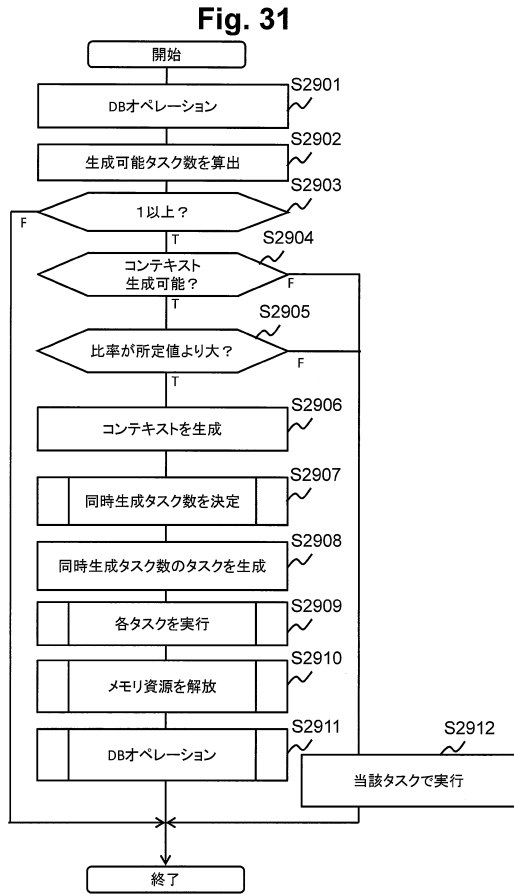


【 図 30 】

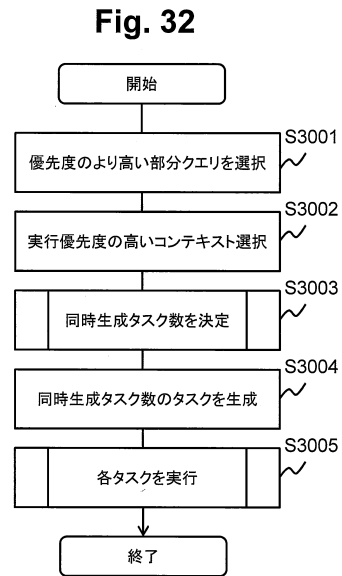
Fig. 30



【 図 3 1 】

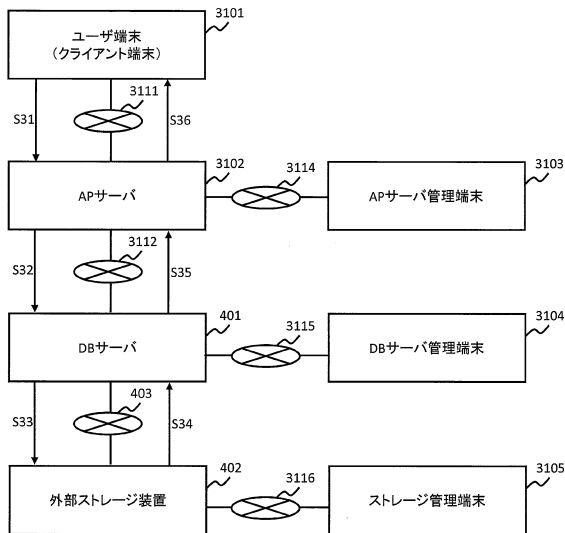


【 図 3 2 】



【 図 3 3 】

Fig. 33



フロントページの続き

- (72)発明者 吉田 美智子
東京都千代田区丸の内一丁目6番6号 株式会社日立製作所内
- (72)発明者 河村 信男
東京都千代田区丸の内一丁目6番6号 株式会社日立製作所内
- (72)発明者 藤原 真二
東京都千代田区丸の内一丁目6番6号 株式会社日立製作所内
- (72)発明者 茂木 和彦
東京都千代田区丸の内一丁目6番6号 株式会社日立製作所内
- (72)発明者 喜連川 優
東京都文京区本郷七丁目3番1号 国立大学法人東京大学内
- (72)発明者 合田 和生
東京都文京区本郷七丁目3番1号 国立大学法人東京大学内

審査官 田中 幸雄

- (56)参考文献 特開2007-34414(JP,A)
特開2011-159107(JP,A)

- (58)調査した分野(Int.Cl., DB名)
G06F 12/00
G06F 9/50