

(19) 日本国特許庁 (JP)

(12) 特 許 公 報 (B2)

(11) 特許番号

特許第6171494号
(P6171494)

(45) 発行日 平成29年8月2日 (2017.8.2)

(24) 登録日 平成29年7月14日 (2017.7.14)

(51) Int. Cl.

F I

G 0 6 F 9/48 (2006.01)

G 0 6 F 9/46 4 5 2 Z

G 0 6 F 9/46 (2006.01)

G 0 6 F 9/46 4 3 0

G 0 6 F 9/46 4 2 0 Z

請求項の数 5 (全 23 頁)

(21) 出願番号 特願2013-74295 (P2013-74295)
 (22) 出願日 平成25年3月29日 (2013.3.29)
 (65) 公開番号 特開2014-199534 (P2014-199534A)
 (43) 公開日 平成26年10月23日 (2014.10.23)
 審査請求日 平成27年12月4日 (2015.12.4)

前置審査

(73) 特許権者 000005223
 富士通株式会社
 神奈川県川崎市中原区上小田中4丁目1番
 1号
 (74) 代理人 100092152
 弁理士 服部 毅巖
 (72) 発明者 古森 良男
 石川県金沢市増泉3丁目4番30号 株式
 会社富士通北陸システムズ内

審査官 漆原 孝治

(56) 参考文献 特開2008-159019 (JP, A)

最終頁に続く

(54) 【発明の名称】 情報処理装置、処理要求プログラム、および処理要求方法

(57) 【特許請求の範囲】

【請求項 1】

複数の装置それぞれで提供されるリソースを更新するトランザクションの実行時に、前記トランザクションに含まれる複数の処理を並列で実行させる、前記複数の装置のうちの少なくとも一部の複数の並列実行装置のうち、処理に要する時間が所定値より長い並列実行装置への処理要求が、他の並列実行装置よりも先に送信されるように、前記複数の並列実行装置への処理要求の送信順を決定する決定手段と、

該送信順で、前記トランザクションの結果を前記複数の並列実行装置それぞれに確定させる処理要求を、前記複数の並列実行装置それぞれに対して送信し、前記複数の並列実行装置に並列に処理を実行させる制御手段と、

を有する情報処理装置。

【請求項 2】

前記決定手段は、前記複数の装置から、処理の成功率が所定の成功率判定値未満であり、かつ処理時間が所定の処理時間判定値未満の装置を特定し、前記複数の装置のうちの、該特定した装置以外の装置を、並列に処理を実行させる前記複数の並列実行装置とすることを特徴とする請求項 1 記載の情報処理装置。

【請求項 3】

前記決定手段は、前記複数の装置における処理の平均の成功率からの、前記複数の装置それぞれの成功率のかい離率を計算し、成功率のかい離率が閾値未満の装置を特定することを特徴とする請求項 2 記載の情報処理装置。

10

20

【請求項 4】

前記制御手段は、前記特定した装置に対して先行して処理要求を送信し、該装置から処理が正常に完了したことを示す応答を受信した後、並列に処理を実行させる前記複数の並列実行装置それぞれに対して処理要求を送信することを特徴とする請求項 2 または 3 記載の情報処理装置。

【請求項 5】

前記制御手段は、トランザクションの 2 フェーズコミットにおけるリソース操作の処理要求を、決定された送信順で送信することを特徴とする請求項 1 乃至 4 のいずれかに記載の情報処理装置。

10

【発明の詳細な説明】**【技術分野】****【0001】**

本発明は、複数の装置に処理要求を送信する情報処理装置、処理要求プログラム、および処理要求方法に関する。

【背景技術】**【0002】**

コンピュータで実行される処理として、トランザクションがある。トランザクションは、複数の処理を 1 つの処理単位としてまとめたものである。トランザクションでは、実行するすべての処理について、すべて成功するか、すべて失敗するかのいずれかになることが保証される。

20

【0003】

トランザクションのなかには、複数のリソースを操作するものがある。ここでリソースとは、データベースやサービスである。またトランザクションで操作されるリソースには、例えば The Open Group の「Distributed Transaction Processing : The XA Specification」の仕様に規定されているリソースマネージャで提供されるリソースも含まれる。

【0004】

複数のリソースを操作するトランザクションでは、それらのリソースの内容の整合性を保つために、例えば 2 フェーズコミットと呼ばれる技術が用いられる。2 フェーズコミットでは、トランザクションを実行するコンピュータは、各リソースを提供するサーバの準備確認を行う。準備確認では、トランザクションを実行するコンピュータは、リソースを提供するサーバに対して、リソースの更新が可能かどうかを問い合わせる。各サーバは、その問い合わせに応じて更新準備を行い、準備が完了すると、準備完了を応答する。トランザクションを実行するコンピュータは、リソースを提供するすべてのサーバからの準備完了の応答を確認すると、各サーバに対して操作反映（コミット）を指示する。各サーバは、操作反映の指示に応じて、リソースの内容を更新する。操作反映が不可能なリソースが 1 つでも存在する場合は、トランザクションを実行するコンピュータは、リソースを管理するすべてのサーバに対して、リソース操作の取り消し（ロールバック）を指示する。

30

【0005】

2 フェーズコミットに関する技術としては、例えば、計算機間の通信ネットワークを経由したメッセージの送信回数が最小となる経路でコミット制御メッセージを送信する技術が考えられている。

40

【先行技術文献】**【特許文献】****【0006】**

【特許文献 1】特開平 6 - 2 5 9 3 8 8 号公報

【発明の概要】**【発明が解決しようとする課題】****【0007】**

しかし、2 フェーズコミットでは、リソースを提供する複数のサーバそれぞれへの、準

50

備確認や操作反映のメッセージを送信する順番について考慮されていない。そのため、非効率的な順番でメッセージが送信され、全体の処理時間を長期化させるおそれがある。なお、２フェーズコミットに限らず、複数の装置に処理要求を送信し、各装置からの応答を待つ技術であれば、効率の悪い順番で処理要求が送信されると、処理時間が長期化する。

【０００８】

１つの側面では、本件は、複数の装置に実行させる処理の効率化を図ることを目的とする。

【課題を解決するための手段】

【０００９】

１つの案では、決定手段と制御手段とを有する情報処理装置が提供される。決定手段は、並列に処理を実行させる複数の装置のうち、処理に要する時間が所定値より長い装置への処理要求が、他の装置よりも先に送信されるように、複数の装置への処理要求の送信順を決定する。制御手段は、該送信順で、複数の装置それぞれに対して処理要求を送信し、複数の装置に並列に処理を実行させる。

【発明の効果】

【００１０】

１態様によれば、複数の装置に処理を効率的に実行させることが可能となる。

【図面の簡単な説明】

【００１１】

【図１】第１の実施の形態に係るシステムの一例を示す図である。

【図２】第２の実施の形態のシステム構成例を示す図である。

【図３】第２の実施の形態に用いるアプリケーションサーバのハードウェアの一構成例を示す図である。

【図４】第２の実施の形態を実現するための機能を示すブロック図である。

【図５】リソース統計情報記憶部のデータ構造の一例を示す図である。

【図６】トランザクションの手順の一例を示すシーケンス図である。

【図７】処理時間によるリソース一覧の並べ替えの一例を示す図である。

【図８】成功率に基づくリソース一覧の並べ替えの第１の例を示す図である。

【図９】成功率に基づくリソース一覧の並べ替えの第２の例を示す図である。

【図１０】順番判定処理の手順の一例を示す図である。

【図１１】コミット処理の第１の例を示す図である。

【図１２】コミット処理の第２の例を示す図である。

【発明を実施するための形態】

【００１２】

以下、本実施の形態について図面を参照して説明する。なお各実施の形態は、矛盾のない範囲で複数の実施の形態を組み合わせる実施することができる。

〔第１の実施の形態〕

図１は、第１の実施の形態に係るシステムの一例を示す図である。第１の実施の形態では、情報処理装置１０がネットワーク４を介して複数の装置１～３に接続されている。情報処理装置１０は、複数の装置１～３それぞれに対して、処理要求を送信し、処理を実行させることができる。情報処理装置１０は、複数の装置１～３に処理を実行させるために、記憶手段１１、決定手段１２、および制御手段１３を有する。

【００１３】

記憶手段１１は、複数の装置１～３の処理に要する時間に関する統計情報１１ａを記憶する。

決定手段１２は、複数の装置１～３のうち、処理に要する時間（処理時間）が所定値より長い装置への処理要求が、他の装置よりも先に送信されるように、処理要求の送信順を決定する。例えば決定手段１２は、統計情報１１ａを参照し、複数の装置１～３それぞれの処理時間を認識する。そして、決定手段１２は、処理に要する時間が長いほど処理要求が先に送信されるように、送信順を決定することができる。また決定手段１２は、処理時

10

20

30

40

50

間が長いほど上位となる複数の等級（ランク）を設け、並列に処理を実行させる複数の装置 1～3 それぞれについて、処理時間に基づいて等級を決定することもできる。等級としては、例えば処理時間が「長い」、「普通」、「短い」の 3 等級とすることができる。そして決定手段 12 は、等級が上位の装置ほど処理要求が先に送信されるように送信順を決定する。また決定手段 12 は、装置によって処理が実行されるごとに、その装置の処理に要する時間に関する統計情報 11a を更新することもできる。

【0014】

制御手段 13 は、決定手段 12 で決定された送信順で、複数の装置 1～3 それぞれに対して処理要求を送信し、複数の装置 1～3 に並列に処理を実行させる。

このようなシステムによれば、情報処理装置 10 の決定手段 12 により、複数の装置 1～3 それぞれに対し、処理時間が長い装置から順に処理要求を送信するように、処理要求の送信順が決定される。例えば統計情報 11a に、各装置 1～3 の処理時間が設定されている。図 1 の例では、装置 1 の処理時間「 t_1 」、装置 2 の処理時間「 t_2 」、装置 3 の処理時間「 t_3 」との関係が、「 $t_2 < t_1 < t_3$ 」である（ t_1 、 t_2 、 t_3 は、正の実数）。この場合、決定手段 12 は、例えば統計情報 11a を参照し、各装置 1～3 の処理時間を比較する。そして決定手段 12 は、処理時間が最も長い装置 3 を、処理要求の送信順の 1 番目に決定し、処理時間が 2 番目に長い装置 1 を、処理要求の送信順の 2 番目に決定し、処理時間が最も短い装置 2 を、処理要求の送信順の 3 番目に決定する。決定手段 12 は、決定した送信順を、制御手段 13 に通知する。

【0015】

制御手段 13 は、決定された送信順で、複数の装置 1～3 それぞれに処理要求を送信する。図 1 の例では、最初に装置 3 に対して処理要求を送信し、次に装置 1 に対して処理要求を送信し、最後に装置 2 に対して処理要求を送信する。なお制御手段 13 は、装置 1～3 に並列に処理を実行させる。すなわち制御手段 13 は、先に送信した処理要求に対する結果の応答を待たずに、次の処理要求を送信する。

【0016】

制御手段 13 が送信した処理要求は、ネットワーク 4 を介して複数の装置 1～3 それぞれに送られる。処理要求を受信した複数の装置 1～3 それぞれは、処理要求に応じて処理を実行する。そして複数の装置 1～3 それぞれは、処理が終了すると、処理結果を情報処理装置 10 に応答する。

【0017】

これにより、情報処理装置 10 は、処理要求を他の送信順で送信した場合に比べて効率的に処理が行われる。その結果、短い時間で、すべての装置 1～3 からの応答を受信することができる。

【0018】

例えば、処理時間の短い装置 2 から順に処理要求を送信した場合、処理要求の送信間隔の分だけ、他の装置 1、3 への処理要求の送信が遅れる。すると、最も処理時間が長い装置 3 への処理要求の送信が遅れ、装置 3 から処理結果の応答を受信するのも遅れることとなる。それに対し、請求項 1 に示すように、最も処理時間が長い装置 3 への処理要求を最初に送信することで、装置 3 からの応答を、可能な限り早く受信することができる。しかも他の装置 1、2 は、装置 3 よりも処理時間が短いため、装置 3 からの応答の送信より前に、装置 1、2 からの応答の送信が期待できる。すなわち装置 3 に最初に処理要求を送信し、処理を迅速に開始させることで、複数の装置 1～3 に対して処理を並列で実施させた場合において、すべての装置 1～3 で処理が完了するまでの所要時間が短縮される。

【0019】

しかも処理要求がネットワーク 4 を介して転送される場合、ネットワーク 4 の負荷状況に応じて、処理要求の転送の遅延時間も変化する。ネットワーク 4 上で通信を中継する装置は、情報を受信した順に転送するため、処理要求を短い時間間隔で連続して送信しても、送信が後の処理要求は、前の処理要求よりも転送の遅延時間が長くなる可能性がある。そこで第 1 の実施の形態では、処理時間が長い装置 3 への処理要求を最初に送信すること

10

20

30

40

50

で、装置 3 への処理要求が他の装置 1, 2 への処理要求よりも優先して転送されるようにし、全体としての処理の遅延を抑止している。

【 0 0 2 0 】

なお決定手段 1 2 は、並列に処理を実行させる複数の装置 1 ~ 3 の処理時間の平均からの、複数の装置 1 ~ 3 それぞれの処理時間のかい離率に基づいて、複数の装置 1 ~ 3 それぞれの等級を決定するようにしてもよい。例えば決定手段 1 2 は、複数の装置 1 ~ 3 の処理時間の平均からの、複数の装置 1 ~ 3 それぞれの処理時間のかい離率を計算し、処理時間のかい離率が閾値より大きい装置を、処理時間が所定値より長い装置とする。この場合、かい離率が閾値となる処理時間が、処理時間の所定値である。処理時間の平均からのかい離率を用いることで、並列で処理を実行する装置の処理時間の相対的な関係により、処理時間の長さの等級を決定できる。その結果、並列に処理を実行させる装置の組み合わせに応じた適切な階級分けを行うことができる。

10

【 0 0 2 1 】

また決定手段 1 2 は、例えば複数の装置 1 ~ 3 それぞれで提供されるリソースを更新するトランザクションの実行時に、送信順を決定することができる。その場合、制御手段 1 3 は、トランザクションの結果を複数の装置 1 ~ 3 それぞれに確定させる処理要求を、複数の装置 1 ~ 3 それぞれに送信する。トランザクションの結果を確定させる処理は、コミットと呼ばれる。コミットを効率的に行うことで、トランザクションの処理の効率化を図ることができる。

20

【 0 0 2 2 】

なおコミットとしては、例えば 2 フェーズコミットの技術を用いることができる。この場合、制御手段 1 3 は、2 フェーズコミットのリソース操作の処理要求（準備確認の処理要求、または操作反映確定の処理要求）を、決定手段 1 2 が決定した送信順に従って送信する。これにより、2 フェーズコミットを効率的に実行することができる。

【 0 0 2 3 】

また決定手段 1 2 は、トランザクションで更新されるリソースを提供する複数の装置 1 ~ 3 のうちの一部のみを、並列に処理を実行させる装置とすることができる。例えば決定手段 1 2 は、処理の成功率が所定の成功率判定値未満であり、かつ処理時間が所定の処理時間判定値未満の装置を特定する。そして決定手段 1 2 は、複数の装置 1 ~ 3 から特定した装置を除いた装置を、並列に処理を実行させる複数の装置に決定する。この場合、制御手段 1 3 は、特定した装置に対して先行して処理要求を送信する。制御手段 1 3 は、先行して処理要求を送信した装置から処理が正常したことを示す応答を受信した後、並列に処理を実行させる複数の装置それぞれに対して処理要求を送信する。これにより、成功率が悪い装置で処理に失敗した場合、他の装置に無駄な処理を行わずにすみ、処理失敗時の処理効率が向上する。なお先行して処理要求を送信する装置は、処理時間が短いため、その装置からの正常の応答を待ってから、他の装置に処理要求を送信しても、処理の遅延は少なくてすむ。

30

【 0 0 2 4 】

また決定手段 1 2 は、トランザクションで更新されるリソースを提供する複数の装置における処理の平均の成功率からの、複数の装置それぞれの成功率のかい離率を計算し、成功率のかい離率が閾値未満の装置を、処理の成功率が所定値未満の装置と判断することができる。このように、複数の装置の処理の平均の成功率からのかい離率を用いることで、複数の装置の処理の成功率の相対的な関係により、処理の成功率が所定値未満の装置を特定できる。その結果、トランザクションの処理を行う装置の組み合わせに応じた適切な装置を、処理の成功率が所定値以下の装置とすることができる。

40

【 0 0 2 5 】

なお図 1 に示す決定手段 1 2、制御手段 1 3 は、例えば情報処理装置 1 0 が有するプロセッサにより実現することができる。また、記憶手段 1 1 は、例えば情報処理装置 1 0 が有するメモリにより実現することができる。

【 0 0 2 6 】

50

また図 1 に示した各要素間を接続する線は通信経路の一部を示すものであり、図示した通信経路以外の通信経路も設定可能である。

〔第 2 の実施の形態〕

次に第 2 の実施の形態について説明する。第 2 の実施の形態は、アプリケーションソフトウェアの実行時に行われるトランザクションを効率的に行うものである。

【0027】

図 2 は、第 2 の実施の形態のシステム構成例を示す図である。アプリケーションサーバ 100 には、ネットワーク 20 を介して、端末装置 30 や、複数のサーバ 210, 220, 230, … が接続されている。アプリケーションサーバ 100 は、アプリケーションソフトウェアを実行する。例えばアプリケーションサーバ 100 は、端末装置 30 から 10

【0028】

サーバ 210, 220, 230, … は、各種のリソースをアプリケーションサーバ 100 に提供する。提供されるリソースは、データベースやその他のサービス機能である。アプリケーションサーバ 100 は、サーバ 210, 220, 230, … で提供されるリソースを利用したトランザクションを行うことができる。アプリケーションソフトウェアに基づいて実行する処理には、例えばサーバ 210, 220, 230, … で提供されるリソースの更新を含むトランザクションがある。

【0029】

アプリケーションサーバ 100 は、複数のリソースを更新するトランザクションを実行する場合、操作対象のリソースを提供する複数のサーバに対して、処理要求を送信する。そしてアプリケーションサーバ 100 は、例えば 2 フェーズコミットによって、トランザクションの結果の整合性を担保する。アプリケーションサーバ 100 は、2 フェーズコミットにおける調整者 (coordinator) であり、操作対象のリソースを管理するサーバは参加者 (cohorts) である。トランザクションを実行する場合、アプリケーションサーバ 100 は、効率的に処理できるように、処理要求の送信先のサーバの順番を決定する。そしてアプリケーションサーバ 100 は、決定した順番で、サーバに処理要求を送信する。これにより、トランザクションの効率化を図ることができる。

【0030】

図 3 は、第 2 の実施の形態に用いるアプリケーションサーバのハードウェアの一構成例を示す図である。アプリケーションサーバ 100 は、プロセッサ 101 によって装置全体が制御されている。プロセッサ 101 には、バス 109 を介してメモリ 102 と複数の周辺機器が接続されている。プロセッサ 101 は、マルチプロセッサであってもよい。プロセッサ 101 は、例えば CPU (Central Processing Unit)、MPU (Micro Processing Unit)、または DSP (Digital Signal Processor) である。プロセッサ 101 の機能の少なくとも一部を、ASIC (Application Specific Integrated Circuit)、PLD (Programmable Logic Device) などの電子回路で実現してもよい。

【0031】

メモリ 102 は、アプリケーションサーバ 100 の主記憶装置として使用される。メモリ 102 には、プロセッサ 101 に実行させる OS (Operating System) のプログラムやアプリケーションプログラムの少なくとも一部が一時的に格納される。また、メモリ 102 には、プロセッサ 101 による処理に必要な各種データが格納される。メモリ 102 としては、例えば RAM (Random Access Memory) などの揮発性の半導体記憶装置が使用される。

【0032】

バス 109 に接続されている周辺機器としては、HDD (Hard Disk Drive) 103、グラフィック処理装置 104、入力インタフェース 105、光学ドライブ装置 106、機器接続インタフェース 107 およびネットワークインタフェース 108 がある。

【0033】

HDD 103 は、内蔵したディスクに対して、磁氣的にデータの書き込みおよび読み出

10

20

30

40

50

しを行う。HDD 103は、アプリケーションサーバ100の補助記憶装置として使用される。HDD 103には、OSのプログラム、アプリケーションプログラム、および各種データが格納される。なお、補助記憶装置としては、フラッシュメモリなどの不揮発性の半導体記憶装置を使用することもできる。

【0034】

グラフィック処理装置104には、モニタ21が接続されている。グラフィック処理装置104は、プロセッサ101からの命令に従って、画像をモニタ21の画面に表示させる。モニタ21としては、CRT (Cathode Ray Tube) を用いた表示装置や液晶表示装置などがある。

【0035】

入力インタフェース105には、キーボード22とマウス23とが接続されている。入力インタフェース105は、キーボード22やマウス23から送られてくる信号をプロセッサ101に送信する。なお、マウス23は、ポインティングデバイスの一例であり、他のポインティングデバイスを使用することもできる。他のポインティングデバイスとしては、タッチパネル、タブレット、タッチパッド、トラックボールなどがある。

【0036】

光学ドライブ装置106は、レーザ光などを利用して、光ディスク24に記録されたデータの読み取りを行う。光ディスク24は、光の反射によって読み取り可能なようにデータが記録された可搬型の記録媒体である。光ディスク24には、DVD (Digital Versatile Disc)、DVD-RAM、CD-ROM (Compact Disc Read Only Memory)、CD-R (Recordable) / RW (ReWritable) などがある。

【0037】

機器接続インタフェース107は、アプリケーションサーバ100に周辺機器を接続するための通信インタフェースである。例えば機器接続インタフェース107には、メモリ装置25やメモリリーダライタ26を接続することができる。メモリ装置25は、機器接続インタフェース107との通信機能を搭載した記録媒体である。メモリリーダライタ26は、メモリカード27へのデータの書き込み、またはメモリカード27からのデータの読み出しを行う装置である。メモリカード27は、カード型の記録媒体である。

【0038】

ネットワークインタフェース108は、ネットワーク20に接続されている。ネットワークインタフェース108は、ネットワーク20を介して、他のコンピュータまたは通信機器との間でデータの送受信を行う。

【0039】

以上のようなハードウェア構成によって、第2の実施の形態の処理機能を実現することができる。なお、第1の実施の形態に示した情報処理装置10も、図3に示したアプリケーションサーバ100と同様のハードウェアにより実現することができる。

【0040】

アプリケーションサーバ100は、例えばコンピュータ読み取り可能な記録媒体に記録されたプログラムを実行することにより、第2の実施の形態の処理機能を実現する。アプリケーションサーバ100に実行させる処理内容を記述したプログラムは、様々な記録媒体に記録しておくことができる。例えば、アプリケーションサーバ100に実行させるプログラムをHDD 103に格納しておくことができる。プロセッサ101は、HDD 103内のプログラムの少なくとも一部をメモリ102にロードし、プログラムを実行する。またアプリケーションサーバ100に実行させるプログラムを、光ディスク24、メモリ装置25、メモリカード27などの可搬型記録媒体に記録しておくこともできる。可搬型記録媒体に格納されたプログラムは、例えばプロセッサ101からの制御により、HDD 103にインストールされた後、実行可能となる。またプロセッサ101が、可搬型記録媒体から直接プログラムを読み出して実行することもできる。

【0041】

図4は、第2の実施の形態を実現するための機能を示すブロック図である。サーバ21

10

20

30

40

50

0 は、リソースとしてデータベースシステム 2 1 1 を有している。データベースシステム 2 1 1 は、サーバ 2 1 0 がソフトウェアを実行することで実現される。データベースシステム 2 1 1 は、データを管理しており、アプリケーションサーバ 1 0 0 からの要求に応じてデータを更新する。

【 0 0 4 2 】

サーバ 2 2 0 は、リソースとしてメッセージングシステム 2 2 1 を有している。メッセージングシステム 2 2 1 は、サーバ 2 2 0 がソフトウェアを実行することで実現される。メッセージングシステム 2 2 1 は、ユーザに通知するメッセージを管理する。例えばメッセージングシステム 2 2 1 は、アプリケーションサーバ 1 0 0 からの要求に応じて、メッセージを更新する。

10

【 0 0 4 3 】

サーバ 2 3 0 は、リソースとしてエンタープライズ情報システム (E I S : Enterprise Information System) 2 3 1 を有している。エンタープライズ情報システム 2 3 1 は、サーバ 2 3 0 がソフトウェアを実行することで実現される。エンタープライズ情報システム 2 3 1 は、企業の業務を行う情報処理システムである。エンタープライズ情報システム 2 3 1 は、アプリケーションサーバ 1 0 0 からの要求に応じて、業務に関する情報を更新する。

【 0 0 4 4 】

アプリケーションサーバ 1 0 0 は、アプリケーション 1 1 0 とトランザクション制御部 1 2 0 とを有する。アプリケーション 1 1 0 は、アプリケーションサーバ 1 0 0 がアプリケーションソフトウェアを実行することで実現される。アプリケーション 1 1 0 は、処理の実行過程で、ネットワーク 2 0 を介して接続されたサーバで提供されるリソースの内容を更新する処理が発生すると、そのリソースのトランザクションによる操作の要求を送信する。

20

【 0 0 4 5 】

トランザクション制御部 1 2 0 は、アプリケーション 1 1 0 からの、トランザクションに関する処理要求に応じて、2 フェーズコミットによるトランザクションを実行する。トランザクションを実行するために、トランザクション制御部 1 2 0 は、処理要求管理部 1 2 1、順番判定部 1 2 2、および通信制御部 1 2 3 を有する。

【 0 0 4 6 】

30

処理要求管理部 1 2 1 は、アプリケーション 1 1 0 からの処理要求を取得し、順番判定部 1 2 2 に、リソースへの要求の送信順の判定を依頼する。処理要求管理部 1 2 1 は、順番判定部 1 2 2 からリソースを提供するサーバへの要求の送信順を受け取ると、その順番でのリソースへの処理要求の送信を、通信制御部 1 2 3 に通知する。そして処理要求管理部 1 2 1 は、通信制御部 1 2 3 から処理結果を受け取ると、その処理結果をアプリケーション 1 1 0 に通知する。

【 0 0 4 7 】

順番判定部 1 2 2 は、リソースを提供するサーバへの要求の送信順を判定する。例えば順番判定部 1 2 2 は、処理に要する時間が長いリソースへの要求ほど、先に送信するように、順番を判定する。順番判定部 1 2 2 は、順番を判定するために、リソース統計情報記憶部 1 2 2 a、リソース統計情報更新部 1 2 2 b、およびリソース統計情報解析部 1 2 2 c を有する。

40

【 0 0 4 8 】

リソース統計情報記憶部 1 2 2 a は、トランザクション中でのリソースの更新に関する統計情報 (リソース統計情報) を記憶する。リソース統計情報には、例えば更新処理の平均の所要時間が含まれる。リソース統計情報記憶部 1 2 2 a としては、例えばメモリ 1 0 2 または HDD 1 0 3 の記憶領域の一部が使用される。

【 0 0 4 9 】

リソース統計情報更新部 1 2 2 b は、リソース統計情報を算出し、リソース統計情報記憶部 1 2 2 a 内のリソース統計情報を更新する。例えばリソース統計情報更新部 1 2 2 b

50

は、トランザクションが実行されるごとに、リソース統計情報を更新する。

【 0 0 5 0 】

リソース統計情報解析部 1 2 2 c は、リソース統計情報記憶部 1 2 2 a に格納されているリソース統計情報を解析し、トランザクションで更新される複数のリソースそれぞれを有するサーバへの、要求の送信順を決定する。リソース統計情報解析部 1 2 2 c は、決定した送信順を、処理要求管理部 1 2 1 に通知する。

【 0 0 5 1 】

通信制御部 1 2 3 は、処理要求管理部 1 2 1 から、トランザクションに関する要求の送信の通知を受け取ると、その要求を、トランザクションで更新されるリソースを有するサーバに送信する。この際、通信制御部 1 2 3 は、処理要求管理部 1 2 1 から指示された順番で要求を送信する。通信制御部 1 2 3 は、要求に対する結果の応答を各サーバから受信すると、処理結果を処理要求管理部 1 2 1 に通知する。

【 0 0 5 2 】

このような機能を有するシステムにより、トランザクションを効率的に実行することができる。なお、図 4 に示した各要素間を接続する線は通信経路の一部を示すものであり、図示した通信経路以外の通信経路も設定可能である。また図 4 に示すリソース統計情報記憶部 1 2 2 a は、図 1 に示す第 1 の実施の形態の記憶手段 1 1 の一例である。図 4 に示す順番判定部 1 2 2 は、図 1 に示す第 1 の実施の形態の決定手段 1 2 の一例である。図 4 に示す通信制御部 1 2 3 は、図 1 に示す第 1 の実施の形態の制御手段 1 3 の一例である。

【 0 0 5 3 】

次にリソース統計情報記憶部 1 2 2 a に格納されるリソース統計情報について詳細に説明する。

図 5 は、リソース統計情報記憶部のデータ構造の一例を示す図である。図 5 の例では、リソース統計情報記憶部 1 2 2 a には、データテーブル形式のリソース統計情報 3 2 が格納されている。リソース統計情報 3 2 には、リソース、場所、トランザクション回数、成功回数、処理時間、および成功率の欄が設けられている。

【 0 0 5 4 】

リソースの欄には、リソースの識別情報が設定される。図 5 の例では、リソースの欄に、リソースの名称が設定されている。

場所の欄には、リソースの提供場所が設定される。リソースの提供場所としては、例えばリソースが操作しているサーバのアドレス、そのサーバ内でのリソースの場所が設定される。サーバ内でのリソースの場所は、例えばディレクトリパスとファイル名とで表される。

【 0 0 5 5 】

トランザクション回数の欄には、トランザクションによってリソースに対して行われた更新処理の回数が設定される。

成功回数の欄には、トランザクションによってリソースに対して行われた更新処理のうち、成功終了した回数が設定される。

【 0 0 5 6 】

処理時間の欄には、トランザクションにおけるリソース更新に関する要求に応じた処理に要した時間の平均が設定される。

成功率の欄には、トランザクションにおけるリソース更新に関する要求に応じた処理の成功率が設定される。

【 0 0 5 7 】

このようなリソース統計情報 3 2 を用いて、トランザクションが効率的に実行される。以下、トランザクションの手順について詳細に説明する。

図 6 は、トランザクションの手順の一例を示すシーケンス図である。まずアプリケーション 1 1 0 が、トランザクション制御部 1 2 0 に対して、トランザクションの開始を通知する（ステップ S 1 1）。トランザクションの開始の通知を受け取ったトランザクション制御部 1 2 0 は、トランザクションの制御を開始する（ステップ S 1 2）。

【 0 0 5 8 】

次にアプリケーション 1 1 0 は、トランザクションにおいて更新する複数のリソースそれぞれに対して、リソース更新操作を行うこと通知する（ステップ S 1 3）。図 6 の例では、データベースシステム 2 1 1、メッセージングシステム 2 2 1、およびエンタープライズ情報システム 2 3 1 が操作対象のリソースであり、これらのリソースに、リソース更新操作を行うことが通知されている。例えばデータベースシステム 2 1 1 に対する S Q L（Structured Query Language）文の発行や、メッセージングシステム 2 2 1 へのメッセージ送受信などが行われる。

【 0 0 5 9 】

このとき、トランザクション制御部 1 2 0 は、アプリケーション 1 1 0 から各リソースへの通知を監視し、どのリソースに対して通知が行われたのかを認識する。そしてトランザクション制御部 1 2 0 は、操作対象のリソースを記憶する（ステップ S 1 4）。

【 0 0 6 0 】

続いて、アプリケーション 1 1 0 は、トランザクション制御部 1 2 0 へ、操作反映を通知する（ステップ S 1 5）。トランザクション制御部 1 2 0 は、アプリケーション 1 1 0 からの操作反映の通知を受け取ると、ステップ S 1 4 で記憶しておいた操作対象のリソース一覧を、順番判定部 1 2 2 へ通知する（ステップ S 1 6）。

【 0 0 6 1 】

順番判定部 1 2 2 は、操作対象のリソース一覧を受信すると、リソース統計情報に基づいて、操作対象のリソースそれぞれの、要求に対する平均処理時間を比較する。また順番判定部 1 2 2 は、処理の成功率に基づいて、各リソースにおける処理の成功率の善し悪しを判断する。そして順番判定部 1 2 2 は、平均処理時間の比較結果と、成功率の善し悪しに基づいて、要求を送信するリソースの順番を判定する（ステップ S 1 7）。例えば順番判定部 1 2 2 は、リソース統計情報記憶部 1 2 2 a 内のリソース統計情報 3 2 を参照し、受信した操作対象のリソース一覧を、処理時間が長いリソースほど上位となるように並べ替える。リソース一覧の上位に登録されたリソースほど、先に要求を送信するリソースであることを示している。また順番判定部 1 2 2 は、処理時間が短い、成功率が悪いリソースがある場合、そのリソースを一覧の最上位に移動し、他のリソースよりも先に要求を送信するように、送信順を判定することもできる。要求の送信順を判定すると、順番判定部 1 2 2 は、判定した順番を示すリソース一覧を、トランザクション制御部 1 2 0 に通知する（ステップ S 1 8）。

【 0 0 6 2 】

トランザクション制御部 1 2 0 は、順番判定部 1 2 2 から要求の送信順を示すリソース一覧を取得する（ステップ S 1 9）。そしてトランザクション制御部 1 2 0 は、取得した一覧の上位のリソースほど処理要求の送信が先となるように送信順を指定して、操作反映依頼を通信制御部 1 2 3 に送信する（ステップ S 2 0）。

【 0 0 6 3 】

通信制御部 1 2 3 は、操作反映依頼に従って、操作反映処理を並列処理で実施する。なお並列処理であっても、要求の送信は順番に行うこととなる。そこで通信制御部 1 2 3 は、操作反映依頼で指定された送信順で、各リソースへ準備確認の要求を送信する（ステップ S 2 1）。

【 0 0 6 4 】

準備確認の要求を受信したデータベースシステム 2 1 1、メッセージングシステム 2 2 1、およびエンタープライズ情報システム 2 3 1 は、2 フェーズコミットにおける操作反映の準備をする。操作反映の準備とは、リソースの更新（コミット）を行える状態まで、処理を進めることである。このとき実施した処理を取り消すことができるように、実施した処理の履歴が、各リソースにおいて記録される。データベースシステム 2 1 1、メッセージングシステム 2 2 1、およびエンタープライズ情報システム 2 3 1 は、それぞれ操作反映の準備が完了すると、通信制御部 1 2 3 へ準備状況を応答する（ステップ S 2 2、S 2 3、S 2 4）。

10

20

30

40

50

【 0 0 6 5 】

通信制御部 1 2 3 は、各リソースからの準備状況の応答を受信し、操作対象のすべてのリソースにおいて操作準備が正常に完了したことを確認する。そして通信制御部 1 2 3 は、操作反映依頼で指定された送信順で、各リソースに操作反映確定の要求を送信する（ステップ S 2 5）。

【 0 0 6 6 】

操作反映確定の要求を受信したデータベースシステム 2 1 1、メッセージングシステム 2 2 1、およびエンタープライズ情報システム 2 3 1 は、2 フェーズコミットにおける操作反映の確定処理（コミット）を実行する。そしてデータベースシステム 2 1 1、メッセージングシステム 2 2 1、およびエンタープライズ情報システム 2 3 1 は、それぞれコミット完了の応答を、通信制御部 1 2 3 へ送信する（ステップ S 2 6，S 2 7，S 2 8）。

10

【 0 0 6 7 】

通信制御部 1 2 3 は、各リソースからコミット完了の応答を受信すると、操作反映処理の結果をトランザクション制御部 1 2 0 に通知する（ステップ S 2 9）。トランザクション制御部 1 2 0 は、操作反映処理の結果を順番判定部 1 2 2 とアプリケーションに 1 1 0 に通知する（ステップ S 3 0）。なお順番判定部 1 2 2 への結果の通知には、例えば処理時間、処理結果などの情報が含まれる。

【 0 0 6 8 】

順番判定部 1 2 2 は、結果の通知内容に応じて、リソース統計情報記憶部 1 2 2 a 内のリソース統計情報 3 2 を更新する（ステップ S 3 1）。例えば順番判定部 1 2 2 のリソース統計情報更新部 1 2 2 b は、操作対象の各リソースのトランザクション回数を 1 ずつカウントアップする。またリソース統計情報更新部 1 2 2 b は、操作反映が成功したリソースの成功回数を、1 ずつカウントアップする。さらにリソース統計情報更新部 1 2 2 b は、今回の操作反映の処理時間に基づいて、各リソースの処理時間を更新する。例えばリソース統計情報更新部 1 2 2 b は、リソースのカウントアップ前のトランザクション回数と処理時間を乗算し、乗算結果に今回の処理時間を加算する。そしてリソース統計情報更新部 1 2 2 b は、加算結果を、カウントアップ後のトランザクション回数で除算する。除算結果が、更新後の処理時間となる。またリソース統計情報更新部 1 2 2 b は、各リソースの成功率を計算する。例えばリソース統計情報更新部 1 2 2 b は、カウントアップ後の成功回数を、カウントアップ後のトランザクション回数で除算する。そしてリソース統計情報更新部 1 2 2 b は、除算結果を、そのリソースの成功率とする。

20

30

【 0 0 6 9 】

アプリケーション 1 1 0 は、トランザクション制御部 1 2 0 から処理結果を受信する（ステップ S 3 2）。アプリケーション 1 1 0 は、通知された処理結果に応じて以降の業務処理を継続する。

【 0 0 7 0 】

このような手順でトランザクションが実行される。そして第 2 の実施の形態では、各リソースへ操作反映を並列処理で実行する際に、高速に処理が実行できるような適切な順番で、リソースに要求を送信する。

【 0 0 7 1 】

高速に処理が実行できる順番は、リソース統計情報解析部 1 2 2 c によって判定される。例えばリソース統計情報解析部 1 2 2 c は、リソース一覧 3 3 に含まれる各リソースの処理時間および成功率を解析し、処理時間が最短となる処理順番を決定する。そしてリソース統計情報解析部 1 2 2 c は、処理順番が早い順となるように、リソース一覧を並べ替える。

40

【 0 0 7 2 】

例えばリソース統計情報解析部 1 2 2 c は、まず、処理時間によってリソース一覧の並べ替えを行う。

図 7 は、処理時間によるリソース一覧の並べ替えの一例を示す図である。例えばリソース統計情報解析部 1 2 2 c は、まず操作対象のリソースの処理時間を評価する。処理時間

50

の評価は、例えば「短い」、「普通」、「長い」の3段階で行われる。3段階で処理時間を評価する場合、「短い」と判定するための閾値（第1の時間閾値）と、「長い」と判定するための閾値（第2の時間閾値）とが設けられる。処理時間が第1の時間閾値より短いリソースは、処理時間が「短い」と評価される。処理時間が第2の時間閾値より長いリソースは、処理時間が「長い」と評価される。処理時間が第1の時間閾値から第2の時間閾値までの範囲に含まれるリソースは、処理時間が「普通」と評価される。

【0073】

処理時間の閾値は、操作対象の各リソースの処理時間の平均値とのかい離率に基づいて設定することもできる。例えばリソース統計情報解析部122cは、操作対象の各リソースの処理時間の平均値を計算する。そしてリソース統計情報解析部122cは、例えば操作対象のリソースの処理時間から平均値を減算し、減算結果を平均値で除算する。リソース統計情報解析部122cは、除算結果を、処理時間のかい離率とする。リソース統計情報解析部122cは、処理時間のかい離率の閾値と、各リソースの処理時間のかい離率とを比較して、処理時間を評価する。

【0074】

処理時間のかい離率の閾値としては、例えば「短い」と判定するための閾値と、「長い」と判定するための閾値との2つを予め設定しておく。また、1つのかい離率の閾値で、処理時間を3段階に評価することもできる。例えばリソース統計情報解析部122cは、リソースの処理時間のかい離率の絶対値が、処理時間のかい離率の閾値より大きく、かつそのリソースの処理時間のかい離率が正の値の場合、処理時間が「長い」と評価する。またリソース統計情報解析部122cは、リソースの処理時間のかい離率の絶対値が、処理時間のかい離率の閾値より大きく、かつそのリソースの処理時間のかい離率が負の値の場合、処理時間が「短い」と評価する。さらにリソース統計情報解析部122cは、リソースの処理時間のかい離率の絶対値が、処理時間のかい離率の閾値以下であれば、処理時間が「普通」とであると評価する。

【0075】

なお、具体的な処理時間のかい離率の閾値は、運用形態やマシンスペックなどに応じて、任意に設定できる。

図7の例では、「リソースB」の処理時間が最も短く、その次に「リソースA」の処理時間が短く、「リソースC」の処理時間が最も長い。リソース統計情報解析部122cは、リソース一覧33に登録されている各リソースを、処理時間が長いほど上位となるようにソートする。すると、ソート後のリソース一覧33では、「リソースC」が最も上位となり、その次に「リソースA」が配置され、最下位に「リソースB」が配置される。

【0076】

ここで、ソート後のリソース一覧33に示される順番を、そのまま要求の送信順とすることができる。また、操作反映処理の成功率に基づいて、順番を入れ替えることもできる。

【0077】

図8は、成功率に基づくリソース一覧の並べ替えの第1の例を示す図である。例えばリソース統計情報解析部122cは、リソース一覧33に含まれる各リソースの成功率を評価する。成功率の評価は、例えば「悪い」か「良い」かの2段階で行われる。2段階で成功率を評価する場合、「悪い」と判定するための閾値（成功率閾値）が設けられる。成功率が成功率閾値より悪い（値が小さい）リソースは、成功率が「悪い」と評価される。成功率が成功率閾値以上のリソースは、成功率が「良い」と評価される。

【0078】

成功率の閾値は、操作対象の各リソースの成功率の平均値とのかい離率に基づいて設定することもできる。例えばリソース統計情報解析部122cは、操作対象の各リソースの成功率の平均値を計算する。そしてリソース統計情報解析部122cは、例えば操作対象のリソースの成功率から平均値を減算し、減算結果を平均値で除算する。リソース統計情報解析部122cは、除算結果を、成功率のかい離率とする。リソース統計情報解析部1

10

20

30

40

50

2 2 c は、成功率のかい離率の閾値と、各リソースの成功率のかい離率とを比較して、成功率を評価する。例えばリソース統計情報解析部 1 2 2 c は、リソースの成功率のかい離率の絶対値が、成功率のかい離率の閾値より大きく、かつそのリソースの成功率のかい離率が負の値の場合、成功率が「悪い」と評価する。またリソース統計情報解析部 1 2 2 c は、リソースの成功率のかい離率の絶対値が、成功率のかい離率の閾値以下である場合、またはリソースの成功率のかい離率の絶対値が正の値の場合、成功率が「良い」と評価する。

【 0 0 7 9 】

なお、具体的な成功率のかい離率の閾値は、運用形態やマシンスペックなどに応じて、任意に設定できる。

10

図 8 の例では、処理時間が短い「リソース B」について成功率が「悪い」と評価され、他のリソースは成功率が良いと評価されている。この場合、リソース統計情報解析部 1 2 2 c は、リソース一覧 3 3 に登録されている各リソースのうち、成功率が「悪い」リソースが最上位となるように並べ替える。

【 0 0 8 0 】

このとき、リソース統計情報解析部 1 2 2 c は、各リソースへの処理（例えば準備確認処理）の処理方法を、リソース一覧 3 3 に設定する。例えば処理時間は短いが成功率は「悪い」リソースの処理の処理方法は「先行実施」に設定される。成功率が「良い」リソースの送信方法は、「並列処理」に設定される。図 8 の例では、「リソース B」の処理方法が「先行実施」であり、その他のリソースの処理方法が「並列処理」である。

20

【 0 0 8 1 】

処理方法「先行実施」とされたリソースは、処理方法「並列処理」のリソースよりも先に処理の要求が送信される。そして、処理方法「先行実施」のリソースの処理が完了するまで、処理方法「並列処理」のリソースへの処理の要求は待たされる。

【 0 0 8 2 】

処理方法「並列処理」とされたリソースは、処理方法「並列処理」の他のリソースと共に、並列に処理される。すなわち「並列処理」のリソースへは、「並列処理」の他のリソースの処理が完了するのを待たずに、処理の要求が送信される。なお「並列処理」の複数のリソースそれぞれへの処理の要求は、リソース一覧 3 3 の上位のリソースから順番に送信される。

30

【 0 0 8 3 】

処理時間が短いリソースの成功率が悪い場合、そのリソースの処理が成功したことを確認後に他のリソースへの処理の要求を送信することで、処理時間が短いリソースでの処理失敗時のロールバックを短時間で行うことができる。ロールバックとは、リソースの状態を、処理の実行前の状態に戻すことである。処理時間が短いリソースでの処理に失敗したとき、他のリソースへは、まだ処理の要求が送信されていない。そのため、他のリソースへの処理要求の送信を抑止でき、通信回数が削減されると共に、処理時間が短縮される。

【 0 0 8 4 】

また、処理時間が長いリソースの成功率が悪い場合もある。この場合の並べ替えの例を図 9 に示す。

40

図 9 は、成功率に基づくリソース一覧の並べ替えの第 2 の例を示す図である。図 9 の例では、処理時間が長い「リソース C」について成功率が「悪い」と評価され、他のリソースは成功率が良いと評価されている。この場合、リソース統計情報解析部 1 2 2 c は、リソース一覧 3 3 に登録されている各リソースの並べ替えを行わない。

【 0 0 8 5 】

このとき、リソース統計情報解析部 1 2 2 c は、各リソースへの処理（例えば準備確認処理）の処理方法を、リソース一覧 3 3 に設定する。例えば成功率に関係なく、すべてのリソースの処理の処理方法が「並列処理」に設定される。

【 0 0 8 6 】

このように処理時間が長いリソースの成功率が悪い場合、そのリソースに処理を先行し

50

て実施させると、他のリソースへ処理の要求を送信するのが遅れてしまい、全体として処理の効率の低下が大きくなる。そのため、処理時間が長いリソースの成功率が悪い場合には、すべてのリソースに並列に処理を実行させる方が適切となる。

【 0 0 8 7 】

なお、成功率が「悪い」リソースがない場合には、要求の送信順は、処理時間に応じた順番のままとなる。その場合の処理方法は、すべてのリソースについて「並列処理」となる。

【 0 0 8 8 】

このような処理の順番の判定処理の手順の詳細を、以下に説明する。

図 10 は、順番判定処理の手順の一例を示す図である。

10

【ステップ S 1 0 1】リソース統計情報解析部 1 2 2 c は、操作対象の各リソースの処理時間のかい離率を計算する。

【 0 0 8 9 】

【ステップ S 1 0 2】リソース統計情報解析部 1 2 2 c は、操作対象の各リソースについて、かい離率に基づいて処理時間を評価する。例えば、かい離率の絶対値が閾値により高く、かい離率が正の値のリソースは、処理時間が「長い」と判断される。また例えば、かい離率の絶対値が閾値により高く、かい離率が負の値のリソースは、処理時間が「短い」と判断される。さらにかい離率が閾値以下のリソースは、処理時間が「普通」と判断される。

【 0 0 9 0 】

20

【ステップ S 1 0 3】リソース統計情報解析部 1 2 2 c は、リソース一覧 3 3 を、処理時間の評価によってソートする。例えば、処理時間が「長い」リソースが最上位、処理時間が「普通」のリソースをその次、処理時間が「短い」リソースが最下位となるように、リソースがソートされる。

【 0 0 9 1 】

【ステップ S 1 0 4】リソース統計情報解析部 1 2 2 c は、成功率の悪いリソースを抽出する。例えばリソース統計情報解析部 1 2 2 c は、リソース一覧 3 3 から、成功率の平均からのかい離率が所定値以上ある、平均より悪い成功率のリソースを抽出する。

【 0 0 9 2 】

【ステップ S 1 0 5】リソース統計情報解析部 1 2 2 c は、成功率の悪いリソースがあったか否かを判断する。成功率の悪いリソースがある場合、処理がステップ S 1 0 6 に進められる。成功率の悪いリソースがなければ、処理がステップ S 1 0 9 に進められる。

30

【 0 0 9 3 】

【ステップ S 1 0 6】成功率が悪いリソースが存在する場合、リソース統計情報解析部 1 2 2 c は、成功率が悪いリソースの処理時間の評価が「短い」か否かを判断する。処理時間の評価が「短い」であれば、処理がステップ S 1 0 7 に進められる。処理時間の評価が「長い」または「普通」であれば、処理がステップ S 1 0 9 に進められる。

【 0 0 9 4 】

【ステップ S 1 0 7】成功率が悪いリソースの処理時間の評価が「短い」である場合、リソース統計情報解析部 1 2 2 c は、抽出したリソースを、リソース一覧 3 3 の最上位に移動する。

40

【 0 0 9 5 】

【ステップ S 1 0 8】リソース統計情報解析部 1 2 2 c は、抽出したリソースの処理方法を「先行処理」に決定し、その処理方法をリソース一覧 3 3 に設定する。またリソース統計情報解析部 1 2 2 c は、抽出したリソース以外のリソースの処理方法を「並列処理」に決定し、その処理方法をリソース一覧 3 3 に設定する。その後、順番判定処理が終了する。

【 0 0 9 6 】

【ステップ S 1 0 9】成功率が悪いリソースがないか、あるいは成功率の悪いリソースの処理時間の評価が「短い」以外の場合、リソース統計情報解析部 1 2 2 c は、すべての

50

リソースの処理方法を「並列処理」に決定する。そして、リソース統計情報解析部 122c は、決定した処理方法を、リソース一覧 33 に設定する。その後、順番判定処理が終了する。

【0097】

このようにして、効率的な処理順が決定される。

図 11 は、コミット処理の第 1 の例を示す図である。図 11 には、処理時間の短いリソースでの処理の成功率が悪くない場合の例が示されている。この場合、処理時間が長い順に、各リソースに、リソース操作処理の要求が送信される。そして各リソースでの処理が、並列に実行される。

【0098】

図 11 の例では、エンタープライズ情報システム 231 の処理時間が最も長く、次にデータベースシステム 211 の処理時間が長く、メッセージングシステム 221 の処理時間が最も短い。この場合、トランザクション制御部 120 は、エンタープライズ情報システム 231、データベースシステム 211、メッセージングシステム 221 の順で、リソース操作処理の要求を送信する。図 11 の例は、2 フェーズコミットでのトランザクションである。そこで、まず準備確認フェーズのリソース操作が以下の手順で行われる。

【0099】

最初に、トランザクション制御部 120 は、準備確認のリソース操作の要求をエンタープライズ情報システム 231 に送信する（ステップ S41）。これにより、エンタープライズ情報システム 231 において、準備確認の処理が実行される（ステップ S42）。次にトランザクション制御部 120 は、準備確認のリソース操作の要求をデータベースシステム 211 に送信する（ステップ S43）。これにより、データベースシステム 211 において、準備確認の処理が実行される（ステップ S44）。さらにトランザクション制御部 120 は、準備確認のリソース操作の要求をメッセージングシステム 221 に送信する（ステップ S45）。これにより、メッセージングシステム 221 において、準備確認の処理が実行される（ステップ S46）。各リソースは、リソース操作の準備が完了すると、準備完了の応答をトランザクション制御部に送信する。

【0100】

トランザクション制御部 120 により、すべてのリソースから準備完了を示す応答を受信したことが確認されると、操作反映フェーズのリソース操作が以下の手順で行われる。

最初に、トランザクション制御部 120 は、操作反映のリソース操作の要求をエンタープライズ情報システム 231 に送信する（ステップ S51）。これにより、エンタープライズ情報システム 231 において、操作反映の処理が実行される（ステップ S52）。次にトランザクション制御部 120 は、操作反映のリソース操作の要求をデータベースシステム 211 に送信する（ステップ S53）。これにより、データベースシステム 211 において、操作反映の処理が実行される（ステップ S54）。さらにトランザクション制御部 120 は、操作反映のリソース操作の要求をメッセージングシステム 221 に送信する（ステップ S55）。これにより、メッセージングシステム 221 において、操作反映の処理が実行される（ステップ S56）。各リソースは、リソース操作の反映が完了すると、反映完了の応答をトランザクション制御部に送信する。

【0101】

図 11 に示したような順番でリソース操作の要求を送信することで、2 フェーズコミットによるトランザクションが効率的に実行される。すなわち、各フェーズの処理時間は、最も処理時間の長いリソースの処理時間と等しくなる。一方、処理時間の短いリソースへ先に処理要求を送信すると、図 11 の例では、メッセージングシステム 221 への処理要求の送信からエンタープライズ情報システム 231 への処理要求の送信までの時間だけ、余分に時間がかかる。

【0102】

しかも、ネットワーク上では、処理要求の転送が出力順に行われるため、処理要求の送信時刻の差が、ネットワーク上での処理要求の転送順に影響を及ぼす。すなわちネットワ

10

20

30

40

50

ークにおいて処理要求を転送する機器（ルータやスイッチ）は、先に送信された処理要求を優先的に転送する。そのためネットワークが混雑していると、トランザクション制御部 120 からの処理要求の送出時間の差よりも、各リソースでの処理要求の受信時間の差が拡大することがある。図 11 に示すように、処理時間が短いリソースに対する処理要求のネットワークでの転送の遅延が多少拡大しても、各フェーズにおける処理時間に対して影響を及ぼさずにすむ。その結果、ネットワークの混雑によるトランザクションの長期化が抑止される。

【0103】

次に、処理時間が短いリソースの処理の成功率が低い場合の例について説明する。

図 12 は、コミット処理の第 2 の例を示す図である。図 12 には、処理時間の短いリソースでの処理の成功率が悪い場合の例が示されている。この場合、処理時間が短いリソースに、リソース操作の要求が先行して送信される。そして、そのリソースの処理の結果に応じたリソース操作の要求が送信される。例えば処理時間が短いリソースの準備確認が正常に完了すれば、他のリソースへの準備確認のリソース操作の要求が送信される。他方、処理時間が短いリソースの準備確認が異常終了すれば、すべてのリソースへのロールバックのリソース操作の要求が送信される。図 12 には、処理時間が短いリソースの準備確認が異常終了した場合の例が示されている。

【0104】

図 12 の例では、エンタープライズ情報システム 231 の処理時間が最も長く、次にデータベースシステム 211 の処理時間が長く、メッセージングシステム 221 の処理時間が最も短い。またメッセージングシステム 221 の処理の成功率は低い。この場合、トランザクション制御部 120 は、最初に準備確認のリソース操作の要求を、メッセージングシステム 221 に送信する（ステップ S61）。これにより、メッセージングシステム 221 において、準備確認の処理が実行される（ステップ S62）。準備確認が異常終了すると、メッセージングシステム 221 は、異常を示す操作結果を、トランザクション制御部 120 に応答する。

【0105】

トランザクション制御部 120 は、準備確認に対する異常の応答を受け取ると、すべてのリソースに対してロールバックのリソース操作の要求を送信する。例えばトランザクション制御部 120 は、ロールバックのリソース操作の要求を、処理時間が長いリソースから順に送信する。図 12 の例では、トランザクション制御部 120 は、まず、エンタープライズ情報システム 231 に送信する（ステップ S71）。これにより、エンタープライズ情報システム 231 において、ロールバックの処理が実行される（ステップ S72）。次に、トランザクション制御部 120 は、データベースシステム 211 に対して、ロールバックのリソース操作の要求を送信する（ステップ S73）。これにより、データベースシステム 211 において、ロールバックの処理が実行される（ステップ S74）。さらにトランザクション制御部 120 は、メッセージングシステム 221 に対して、ロールバックのリソース操作の要求を送信する（ステップ S75）。これにより、メッセージングシステム 221 において、ロールバック処理が実行される（ステップ S76）。

【0106】

このように、成功率が低く、かつ処理時間が短いメッセージングシステム 221 に対するリソース操作を先行実施することで、操作が異常終了した場合のロールバックを迅速に行うことができる。すなわち、図 11 に示したように、メッセージングシステム 221 へのリソース操作の処理要求の送信を最後に行うと、メッセージングシステム 221 から異常の応答を受信する時期が遅れる。その結果、ロールバックを開始する時期が遅れてしまう。図 12 に示すように、メッセージングシステム 221 へのリソース操作を先行実施すれば、早期に異常終了の応答を受信し、ロールバックを開始できる。しかも、メッセージングシステム 221 の処理時間は短いため、トランザクション全体の処理の長期化は、最小限に抑えることができる。

【0107】

以上説明したように、第2の実施の形態によれば、処理的に処理できる順番で、処理の要求を送信するようにしたため、トランザクション全体の効率化を図ることができる。しかも第2の実施の形態では、処理時間が短いリソースでの成功率が低い場合、そのリソースへの処理要求を先行して実施し、そのリソースでの処理が正常に完了したことを確認した後に、他のリソースへの処理要求を送信するようにしている。これにより、処理の成功率の低いリソースでの処理が異常終了した場合のロールバックを迅速に行うことができる。

【0108】

〔その他の実施の形態〕

第2の実施の形態では、トランザクションが行われるごとに、リソース統計情報を更新している。この場合、システムの運用開始後の初期段階では、リソース統計情報算出の元となるトランザクションの数が少なく、統計的な信頼性に乏しくなる。そこで、リソース統計情報の初期設定として、数パターンのデフォルトの統計情報雛型を容易することができる。例えば、以下の手順でリソース統計情報の初期設定をすることができる。

【0109】

〔第1のステップ〕システムの管理者は、リソース統計情報の複数の雛形から、使用する雛形を決定する。

例えば、以下の観点でデフォルトのリソース統計情報の雛型を数パターン用意される。

- ・リソース数（参加リソース数の少ない、多い）
- ・ネットワーク環境（回線速度、回線品質）
- ・マシンスペック（高スペック、低スペック）

管理者は、数パターンの雛型より適用する環境に最も近い雛型を選択する。雛形には、例えばリソース名や場所が空欄の状態、リソースの処理時間や成功率が予め設定されている。また雛形には、所定量のトランザクション回数が設定されている。例えば適用するシステムと同程度の規模のシステムにおいて、所定期間（例えば1日）に処理するトランザクション回数が設定される。雛形の成功回数は、トランザクション回数と成功率から算出した値である。

【0110】

〔第2のステップ〕システムの管理者は、適用するシステムのリソースの情報（リソース名や場所）を、決定した雛形に入力する。この際、管理者は、処理時間が遅いと想定される以下のリソースの優先順位が高くなるように、リソース情報を登録する。

- ・回線速度が遅いネットワークに配置されるリソース
- ・マシンスペックが低いマシンに配置されるリソース

〔第3のステップ〕システムの管理者は、処理時間の評価における「長い／短い」の判断の閾値を変更する。例えば、管理者は以下の考え方で閾値を変更する。

- ・各リソースの処理時間のバラツキが小さい場合は、閾値を小さくする。
- ・各リソースの処理時間のバラツキが大きい場合は、閾値を大きくする。

【0111】

このような閾値の変更は、例えば管理者がアプリケーションサーバ100に、「長い／短い」の判断の閾値を入力することで実施される。なおアプリケーションサーバ100には、処理時間の評価における「長い／短い」の判断の閾値の初期値（デフォルト値）を予め設定しておくことができる。閾値として初期値を使用できる場合、管理者は、閾値の変更処理を行わなくてもよい。

【0112】

〔第4のステップ〕システムの管理者は、成功率の評価における「良い／悪い」の判断の閾値を変更する。例えば管理者は、以下の考え方で閾値を変更する。

- ・各リソースの成功率が良い場合は、閾値を小さくする。
- ・各リソースの成功率が悪い場合は、閾値を大きくする。

【0113】

このような閾値の変更は、例えば管理者がアプリケーションサーバ100に、「良い／

10

20

30

40

50

悪い」の判断の閾値を入力することで実施される。なおアプリケーションサーバ 100 には、成功率の評価における「良い／悪い」の判断の閾値の初期値（デフォルト値）を予め設定しておくことができる。閾値として初期値を使用できる場合、管理者は、閾値の変更処理を行わなくてもよい。

【0114】

〔第5のステップ〕システムの管理者は、編集した雛形を、リソース統計情報 32 の初期値として適用する。例えばシステムの管理者は、編集後の雛形のリソース統計情報記憶部 122a への格納指示を、アプリケーションサーバ 100 に入力する。するとアプリケーションサーバ 100 において、編集後の雛形が、リソース統計情報 32 としてリソース統計情報記憶部 122a に格納される。そして、システムの運用が開始される。

10

【0115】

このように、雛形を用いてリソース統計情報の初期値を設定できるようにすることで、以下の利点がある。

- ・運用の当初から、効率的なトランザクションを実施できる。
- ・処理順番の決定における精度向上までの時間を短縮できる。

【0116】

また、リソース統計情報を不揮発化することもできる。不揮発化とは、不揮発性の記録媒体に保存することである。

リソース統計情報を不揮発化することで、以下の利点がある。

- ・予期しないシステムダウンによるリソース統計情報の損失を防止できる。
- ・不揮発化することでリソース統計情報のマシン間における持ち運びが可能となり、適用環境からクローン環境へ、または試験環境から運用環境へなどの持ち運びでリソース統計情報の流用が可能になる。

20

【0117】

リソース統計情報の不揮発化方法としては、例えば、トランザクション制御部 120 がメモリ 102 上に展開しているリソース統計情報 32 を、入出力装置である HDD 103 にコピーする。不揮発化を行うタイミングは任意に指定可能である。例えば不揮発化のタイミングとして、以下のようにしてタイミングを指定可能である。

- ・不揮発化の実施時刻を指定する。
- ・運用停止時に不揮発化を実施することを指定する。
- ・通信時における処理時間の「短い／長い」の閾値を超過するリソースの検出時に不揮発化を実施することを指定する。
- ・処理時間の「短い／長い」の閾値の超過回数が所定値を超えた場合に、不揮発化を実施することを指定する。
- ・処理結果における成功率の「良い／悪い」の閾値を超過するリソース検出時に不揮発化を実施することを指定する。
- ・処理結果における成功率の「良い／悪い」の閾値の超過回数が所定値を超えた場合に、不揮発化を実施することを指定する。

30

【0118】

以上、実施の形態を例示したが、実施の形態で示した各部の構成は同様の機能を有する他のものに置換することができる。また、他の任意の構成物や工程が付加されてもよい。さらに、前述した実施の形態のうちの任意の 2 以上の構成（特徴）を組み合わせたものであってもよい。

40

【符号の説明】

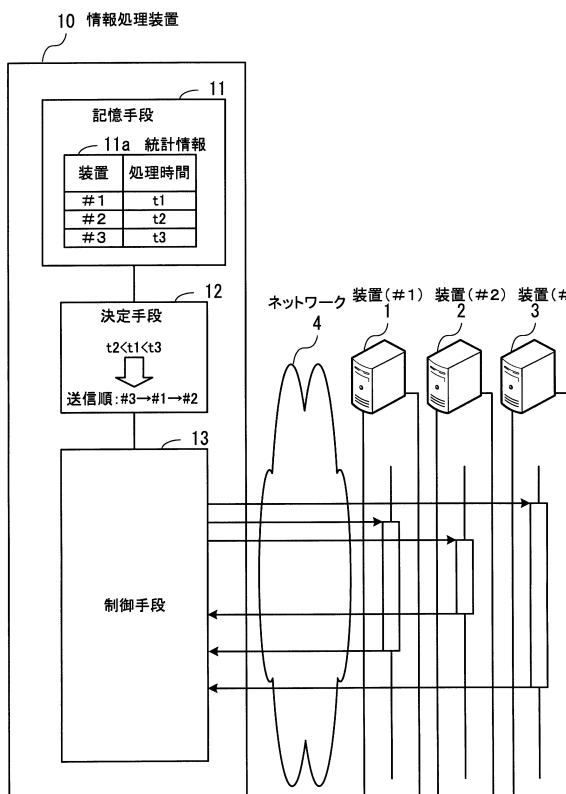
【0119】

- 1～3 装置
- 4 ネットワーク
- 10 情報処理装置
- 11 記憶手段
- 11a 統計情報

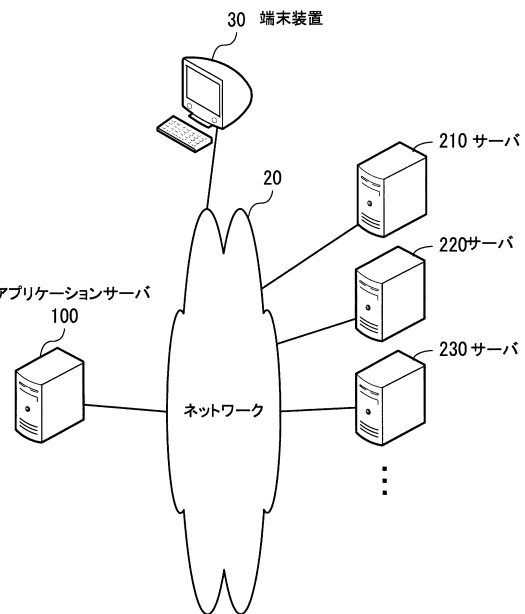
50

- 1 2 決定手段
1 3 制御手段

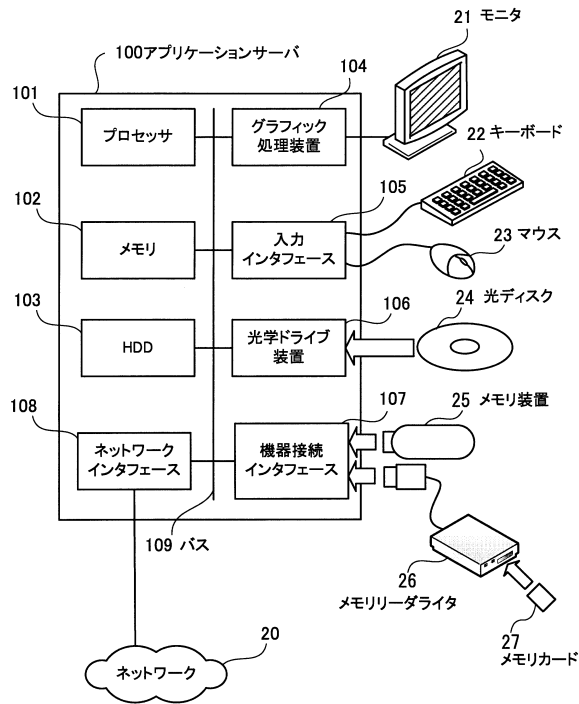
【図 1】



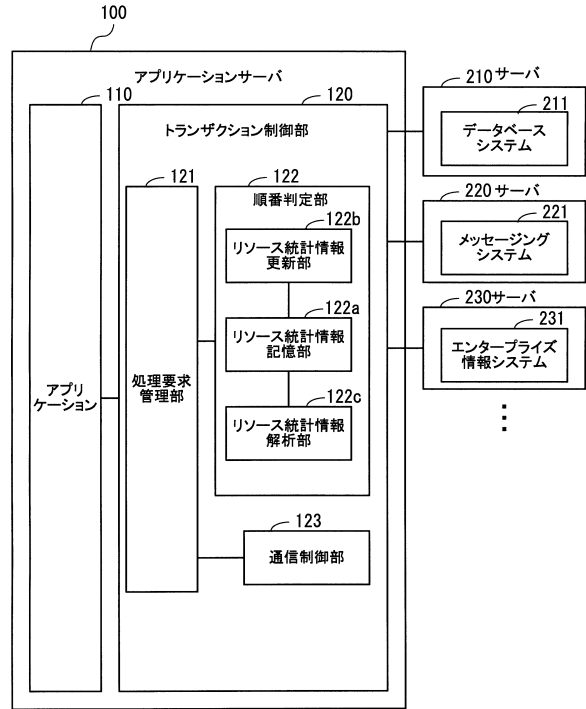
【図 2】



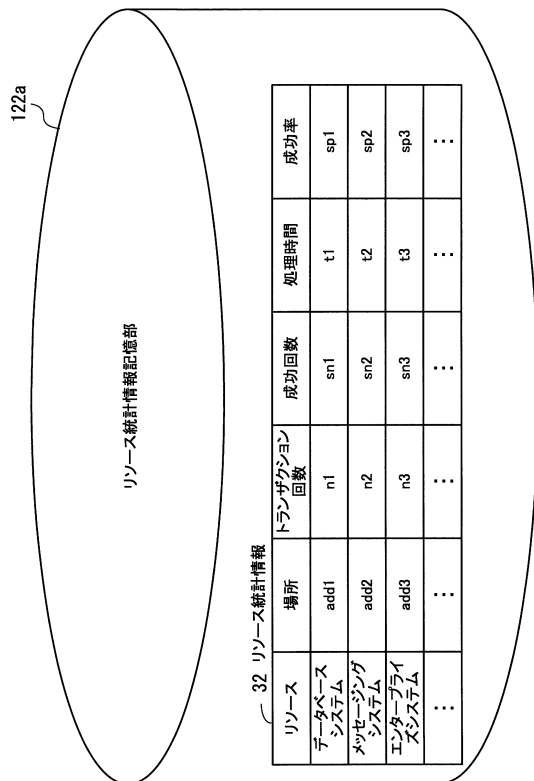
【図 3】



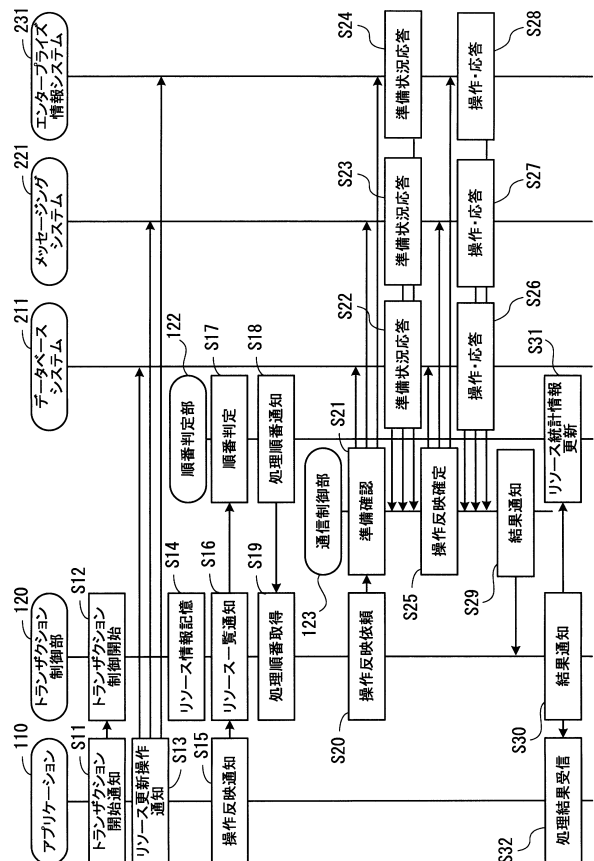
【図 4】



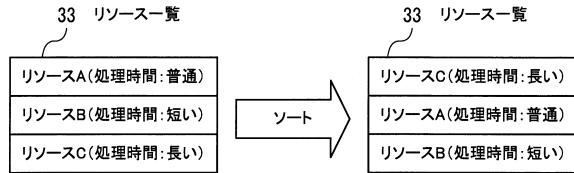
【図 5】



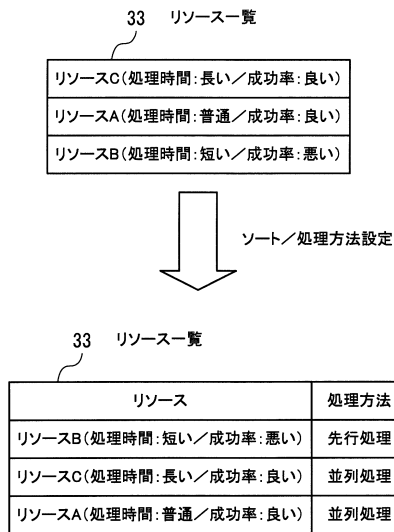
【図 6】



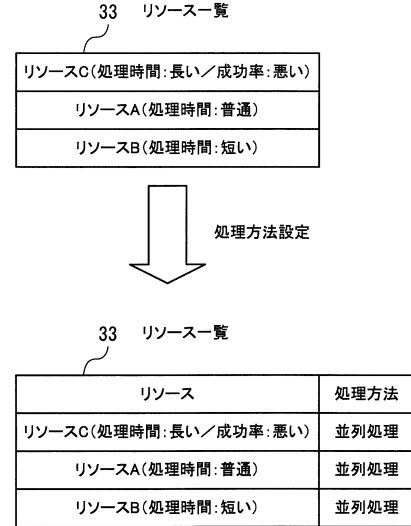
【図 7】



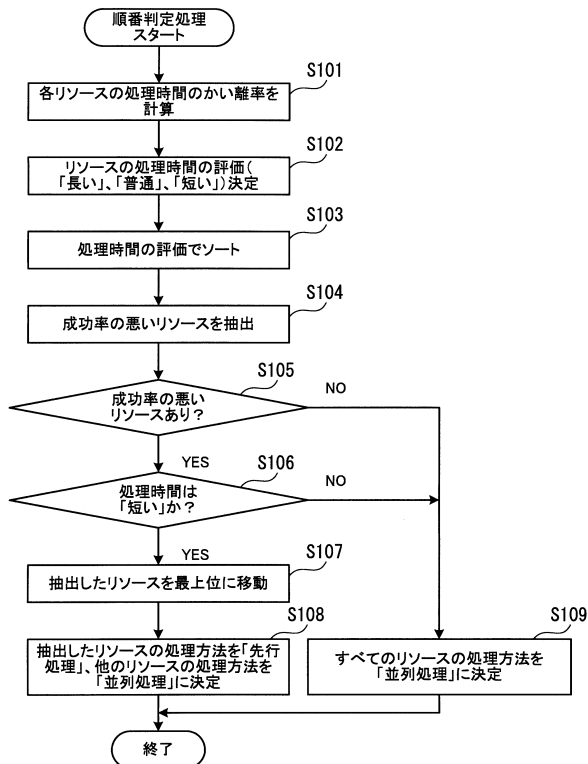
【図 8】



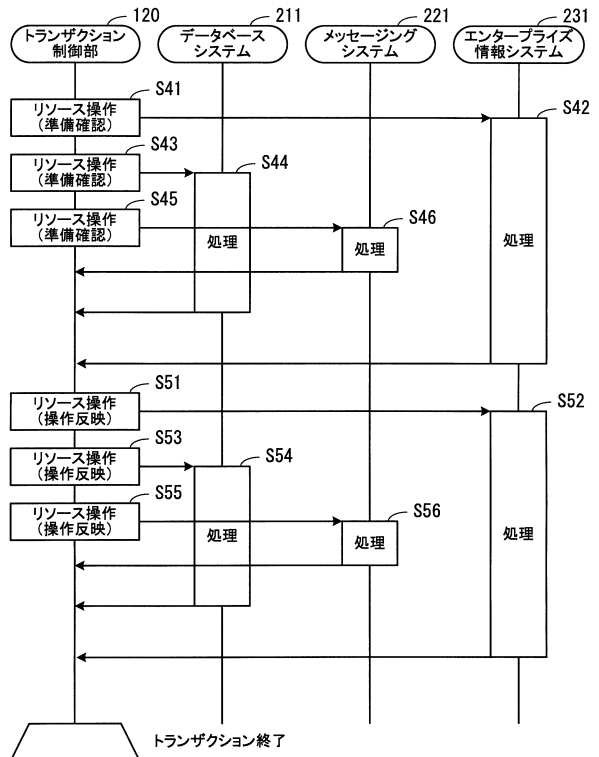
【図 9】



【図 10】



【図 11】



```

sequenceDiagram
    participant 120 as トランザクション制御部
    participant 211 as データベースシステム
    participant 221 as メッセージングシステム
    participant 231 as エンタープライズ制御システム

    120->>211: リソース操作 (準備確認) S61
    211-->>120: 操作結果: 異常
    211->>211: 処理 S62
    120->>231: リソース操作 (ロールバック) S71
    120->>211: リソース操作 (ロールバック) S73
    211->>211: 処理 S74
    120->>211: リソース操作 (ロールバック) S75
    211->>211: 処理 S76
    120->>231: リソース操作 (ロールバック) S77
    231->>231: 処理 S72
    120->>120: トランザクション終了
  
```

フロントページの続き

(58)調査した分野(Int.Cl. , D B 名)

G 0 6 F	9 / 4 8
G 0 6 F	9 / 4 6