

明 細 書

発明の名称：

データ管理システム、サービス提供システム及びその機能拡張方法

技術分野

[0001] この発明は、データ管理システム、サービス提供システム及びその機能拡張方法に係り、特に、多数のクライアント端末に対して情報処理サービスを提供するAPサーバの業務処理によって生成されたデータを、複数のDBサーバで重複管理する技術、一つの筐体内に業務処理を担当するAPサーバ機能とデータベースの管理を担当するDBサーバ機能を併存させる技術に関する。

背景技術

[0002] (1) 地震や津波、噴火等の大規模災害、あるいは原発事故やテロ等の人災によってデータベースが物理的に損壊する場合に備え、企業の基幹システムにおいては、業務データを遠隔地に設置したバックアップ用のデータベースに重複登録することが以前より行われている。

例えば、特許文献1においては、正制御装置のディスク装置にデータを登録した後、遠隔地に設置された副制御装置に同一データを送信し、その登録を求める技術が開示されている。

[0003] このように、正副2台の制御装置によってデータを二重持ちすることにより、一方の地域において地震等が発生し、ディスク装置が物理的に破損した場合であっても、生き残った他方のディスク装置内のデータを活用することで、業務処理の連続性を担保することが可能となる。

また、データベースの損壊は地震等によってデータセンター単位で被災する場合に限られるものではなく、DBサーバのマシントラブルによっても生じ得るため、同一構内においてもこのようなバックアップ体制を構築しておくことが望まれる。

[0004] ところで、複数のデータベースに対する更新処理を実行する際には、全てのデータベースに対するコミットが成立しないとデータ相互間に矛盾が生じ

るため、通常はXAインターフェースの2相コミットと称する調停方式が採用され、データベース相互間の整合性が担保される。

[0005] この2相コミットではまず、処理全体を司るマスターサーバが、対象となる各DBサーバに対し、コミットが実行可能であることを問い合わせる。更新準備が整っているDBサーバは「準備完了」の応答を返し、すべてのDBサーバが準備を終えたことを確認した上で、マスターサーバはコミット開始を通知し、データベースが一斉に書き換えられる。書き換え中にいずれかのデータベースで異常が発生した場合、異常が生じたDBサーバは失敗を伝え、マスターサーバはすべてのサーバに処理撤回を通知して、データを復元する「ロールバック」処理を行うよう指示する（非特許文献2参照）。

[0006] このような2相コミットを採用することにより、DBサーバ間でのデータ整合性が確保され、不正なデータが発生することは確かに回避できる。

しかしながら、マスターサーバと遠隔地に設置されたDBサーバ間で何度もデータのやり取りを繰り返す必要があることから、処理の迅速性が犠牲とならざるを得なかった。このため、ATM取引のように数秒程度の待ち時間が生じても問題にならない業務処理に対しては有効であっても、例えばオンライントレードのように、コンマ何秒かの遅延も許されないような業務処理には適用できないという問題があった。

[0007] (2) また、通常、業務処理を担当するAPサーバと、業務処理の結果である各種データをデータベースに格納して管理するDBサーバとは別個のサーバコンピュータで構成され、それぞれの間を通信ネットワーク経由で接続することが行われている。このため、APサーバとDBサーバ間ではイーサネット（登録商標）による通信のオーバーヘッドが発生し、高速な処理が実現できなかった（非特許文献3参照）。

[0008] もちろん、1台のサーバコンピュータ内にAPサーバ用プログラムとDBサーバ用プログラムをセットアップすることにより、APサーバとDBサーバを同一筐体内に形成することは従来でも可能であり、これにより通信のオーバーヘッドをなくすこと自体は実現できる。

しかしながら、この場合にはAPサーバとDBサーバとの間でコンピュータのリソースを奪い合う形になるため、通信のオーバーヘッドが解消されることによる高速化のメリットが半ば相殺されてしまい、全体としての処理速度が劇的に向上することは期待できなかった。

[0009] ところで、コンピュータの処理能力の向上は、以前は主としてCPUのクロック数の向上に依存してきたが、発熱や消費電力の問題が浮上したため、2003年～2005年頃を境にCPUのクロック数は頭打ちと状態となっている。その代わりに、近年ではCPUのマルチコア化が進み、比較的低いクロック数のCPUコアであっても、一つのパッケージ内に4～6個のCPUコアを格納させて処理の並列化を図ることにより、増え続けるデータ処理量に対応することが期待されている。

[0010] 確かに、DNA解析や気象解析のように、純粋な演算処理を主体とした科学技術計算の場合には、多数のCPUコアを連携させることにより、処理能力を比較的容易に向上させることができる。

これに対し、データベースに蓄積された大量のデータを複雑に組み合わせて参照する必要のある業務系処理の場合には、DBサーバにおけるデータの抽出処理や演算処理に時間を要するため、例えAPサーバ及びDBサーバを載せるサーバコンピュータのCPUコア数を増やしたとしても、直ちに処理速度の向上には結びつかないことが指摘されていた。

[0011] (3) また、複数のDBサーバとAPサーバとの連携を前提とする大規模なサービス提供システムにおいて、途中で機能の拡張を行う必要が生じた場合には、各DBサーバに格納されたテーブルに新機能に対応したカラムを追加する修正を施すと共に、APサーバ側のアプリケーションプログラムにも大幅な改変を施す必要があり、その過程でバグが生じやすいという問題があった。

[0012] すなわち、DBサーバに格納された業務データを利用するに際し、通常はAPサーバからDBサーバにSQL文を発行し、DBサーバにジョインやマッチング、ソート等の演算処理を依頼することとなるが、このためには業務処理用のアプリケーションプログラム中にSQL文を記述しておく必要がある（非特許文献4

参照)。

[0013] このSQL文は、一般に複雑かつ長大な記述になるため第三者が内容を理解するのは容易でなく、しかもコンパイルチェックが効かないという問題がある。

また、一旦、各DBサーバに格納されたテーブルに修正を施してしまうと、何らかの不具合によって旧来のサービスに戻す必要が生じた場合に、大きな手間が発生する。

このため、ユーザの要望に迅速に対応したいサービスの運用部門と、システムの安全性を重視する開発部門との間で意見の対立が生じ、頻繁な機能の拡張が阻害されていた。

[0014] 特許文献1：特開平11-85408号

非特許文献1：Webシステム入門／第1回 Webサイトの構成とJ2EEサーバ

インターネットURL：<http://www.atmarkit.co.jp/fjava/rensai2/websys01/websys01.html> 検索日：2014年1月30日

非特許文献2：2相コミット インターネットURL：<http://e-words.jp/w/2E79BB8E382B3E3839FE38383E38388.html> 検索日：2014年1月30日

非特許文献3：「ちょっと待った!」、オーバーヘッドを忘れていませんか
インターネットURL：<http://itpro.nikkeibp.co.jp/free/TIS/denwa/20040517/1/> 検索日：2014年4月3日

非特許文献4：プログラムとデータベースの関係 インターネットURL：http://www.limy.org/program/coding/program_db.html 検索日：2014年7月15日

発明の開示

発明が解決しようとする課題

[0015] この発明は、従来の上記(1)の問題を解決するために案出されたものであり、処理の迅速性を犠牲にすることなく、大切な業務データを地理的に離れた複数拠点に確実に保全できる技術の実現を第1の目的としている。

[0016] また、この発明は、従来の上記(2)の問題を解決するために案出されたもの

であり、APサーバとDBサーバを同一筐体内に実現するに際して、CPUのマルチコアを有効に活用することにより、APサーバ及びDBサーバの処理速度を飛躍的に向上させる技術を提供することを第2の目的としている。

[0017] また、この発明は、従来の上記(3)の問題を解決するために案出されたものであり、複数のDBサーバによって同一の業務データが冗長管理されている環境下において、比較的容易にサービス提供システムの機能を拡張可能とする技術の提供を第3の目的としている。

課題を解決するための手段

[0018] 上記第1の目的を達成するため、請求項1に記載したデータ管理システムは、相互に所定の距離を隔てて設けられた複数のデータセンター内に、それぞれAPサーバと複数のDBサーバを設置したデータ管理システムであって、上記の全DBサーバは、相互に共通するデータを格納するテーブルをそれぞれ備える共に、各テーブルにはレコードの削除及び更新が禁止される制約が設けられており、上記の各APサーバは、同一データセンター内に設置された各DBサーバと接続されると共に、他のデータセンター内に設置された各DBサーバとも接続されており、また各APサーバは、少なくとも一つの業務処理部と、人工キー管理部と、データ制御部と、複数のDB連絡部を備えており、上記業務処理部は、業務処理の結果である業務データを生成するに際し、上記人工キー管理部に対して人工キーの発行を要求すると共に、この人工キー管理部によって発行されたユニークな人工キーよりなるIDを主キーとして備えた業務データを生成し、上記人工キー管理部は、同一データセンター内に設置された人工キー管理テーブルを参照し、現時点で最新の人工キーの値を上記業務処理部に渡すと同時に、当該テーブルの値を更新させる処理を実行し、上記データ制御部は、上記業務処理部から出力された業務データを複製して各DB連絡部に出力すると共に、各DB連絡部を介して各DBサーバからの受取完了通知が送信された際に、全DBサーバからの受取完了通知が揃っていることを確認した上で、上記業務処理部に登録完了通知を出力し、上記の各DB連絡部には、それぞれ担当するDBサーバの特定情報が設定されており、各DB連絡部は

、上記データ制御部から業務データが出力された際に、自己の担当するDBサーバにSQLを発行してレコードの追加を求めると共に、当該DBサーバから受取完了通知が送信された場合には、これを上記データ制御部に出力する処理を実行し、さらに、上記の各人工キーの所定の位置には、APサーバの識別符号が設定されており、上記業務処理部は、上記データ制御部から登録完了通知が渡されるまでは当該業務データの参照を控えることを特徴としている。

[0019] 請求項2に記載したデータ管理システムは、請求項1のシステムであって、さらに上記業務データが、上記業務処理部による業務処理の結果として生成された、相互に関連性を有する複数の実体データと、各データを束ねるために生成された統合データの組合せよりなり、この統合データは、人工キーよりなる統合IDを主キーとして格納するデータ項目のみを備え、上記の各実体データは、当該実体データを格納するデータ項目と、人工キーよりなる独自のIDを主キーとして格納するデータ項目と、上記統合データのIDを外部キーとして格納するデータ項目を備えていることを特徴としている。

[0020] 請求項3に記載したデータ管理システムは、請求項1または2のシステムであって、さらに、上記の各APサーバと各DBサーバとの間が、それぞれ複数のコネクションを介して接続されると共に、各APサーバにはコネクション単位で上記のDB連絡部が設けられており、上記業務データが、各DB連絡部及びコネクションを介して並列的に各DBサーバに送信されることを特徴としている。

[0021] 請求項4に記載したデータ管理システムは、請求項1～3のシステムであって、さらに上記業務処理部が、既存の業務データを更新する必要がある場合に、当該業務データのIDを格納するデータ項目と、人工キーからなる独自のIDを主キーとして格納するデータ項目と、修正後の実体データを格納するデータ項目を備えた業務データを新たに生成し、上記データ制御部及び各DB連絡部を介してこの業務データを各DBサーバに設けられた既存のテーブルに追加登録させることにより、既存の業務データを実質的に更新することを特徴としている。

[0022] 請求項5に記載したデータ管理システムは、請求項1～4のシステムであって、さらに上記業務処理部が、既存の業務データを削除する必要がある場合に、当該業務データのIDを格納するデータ項目と、人工キーからなる独自のIDを主キーとして格納するデータ項目を備えたデータ取消用の業務データを生成し、上記データ制御部及び各DB連絡部を介してこのデータ取消用の業務データを各DBサーバに設けられた取消専用のテーブルに追加登録させることにより、既存の業務データを実質的に削除することを特徴としている。

[0023] 上記第2の目的を達成するため、請求項6に記載したデータ管理システムは、複数のCPUコアを内蔵したCPUパッケージ、及び複数のテーブルを格納しておく外部記憶装置を備えたコンピュータと、このコンピュータ上でAPサーバ用プログラムを起動させることによって構成されたAPサーバと、上記コンピュータ上でDBサーバ用プログラムを複数起動させることによって構成された複数のDBサーバを備え、上記の外部記憶装置には、相互に共通するデータを格納する複数のテーブルが、DBサーバ単位で別個に格納されており、上記のAPサーバ及び各DBサーバには、それぞれ複数の異なるCPUコアが専用的に割り当てられており、上記APサーバは、業務処理の実行に必要なデータの種別を特定する機能と、上記DBサーバの何れかに対して必要なデータの抽出を依頼する機能と、当該DBサーバから取得したデータに対して必要な演算処理を施す機能と、業務処理によって生じたデータの登録を、上記の全DBサーバに対して同時に依頼する機能を少なくとも備えており、上記DBサーバにデータの抽出または登録を依頼するに際しては、当該処理を担当するCPUコアを具体的に指定した状態でSQLを発行し、これを受けたDBサーバでは、指定されたCPUコアが対応の処理を実行することを特徴としている。

[0024] 請求項7に記載したデータ管理システムは、請求項6のシステムであって、さらに、上記コンピュータは上記CPUパッケージを複数備えており、各CPUパッケージには専用のローカルメモリが割り当てられており、上記APサーバ及び上記DBサーバの少なくとも一部には、それぞれ別個のCPUパッケージに属するCPUコアが割り当てられており、上記APサーバ及び各DBサーバには、ロー

カルメモリにしかアクセスできない制約が課せられていることを特徴としている。

[0025] 請求項 8 に記載したデータ管理システムは、請求項 6 または 7 のシステムであって、さらに、上記 AP サーバには、上記の各 DB サーバに対してデータの挿入及び抽出を依頼することのみが許容され、データの更新及び削除を依頼することが禁止される制約が課せられていることを特徴としている。

[0026] 上記第 3 の目的を達成するため、請求項 9 に記載したサービス提供システムは、AP サーバと、この AP サーバと接続された複数の DB サーバとを備え、クライアント端末に対して所定の情報処理サービスを提供するサービス提供システムであって、上記 AP サーバは、上記の各 DB サーバ単位で設けられた複数の DB 連絡部と、業務処理部と、DB 制御部とを備え、上記の各 DB 連絡部には、それぞれ担当する DB サーバの特定情報が設定されており、上記の各 DB サーバには、相互に共通するデータを格納するテーブルがそれぞれ設けられており、各テーブルには、レコードの削除及び更新が禁止される制約が設けられており、上記業務処理部からデータの登録依頼を受けた際に、上記 DB 制御部は、当該データを複製して上記の全 DB 連絡部に渡し、これを受けた各 DB 連絡部は、自己の担当する DB サーバに SQL 文を発行してレコードの追加を求め、上記業務処理部から特定データの参照を求められた場合に、上記 DB 制御部は、上記の各 DB サーバの中から 1 または複数の DB サーバを特定し、当該 DB サーバを担当している DB 連絡部に対してデータの抽出を依頼し、これを受けた DB 連絡部は、自己の担当している DB サーバに対して SQL 文を発行してレコードの抽出を依頼すると共に、DB サーバから受け取ったデータを DB 制御部に渡し、DB 制御部は、受け取ったデータに必要な演算処理を施した上で、上記業務処理部に渡すことを特徴としている。

[0027] 請求項 10 に記載したサービス提供システムは、請求項 9 のシステムであって、さらに、上記業務処理部は、既存データを更新する必要がある場合に、更新後の値を備えたデータを新たに生成し、上記 DB 制御部及び各 DB 連絡部を介してこのデータを各 DB サーバに設けられた既存のテーブルに追加登録さ

せることにより、既存データの実質的な更新を実現することを特徴としている。

[0028] 請求項 1 1 に記載したサービス提供システムは、請求項 9 または 1 0 のシステムであって、さらに、上記業務処理部は、既存の業務データを削除する必要がある場合に、当該業務データの ID を格納するデータ項目を備えたデータ取消用のデータを生成し、上記 DB 制御部及び各 DB 連絡部を介してこのデータ取消用のデータを各 DB サーバに設けられた取消専用のテーブルに追加登録させることにより、既存データの実質的な削除を実現することを特徴としている。

[0029] 請求項 1 2 に記載したサービス提供システムは、請求項 9 ～ 1 1 のシステムであって、さらに、上記 AP サーバ及び複数の DB サーバの中の少なくとも一部が、同一のコンピュータ上に形成されており、このコンピュータは、複数の CPU コアを内蔵した CPU パッケージと、複数のテーブルを格納しておく記憶装置を備え、このコンピュータ上で AP サーバ用プログラムを起動させることによって上記 AP サーバが構成されると共に、この AP サーバ上で専用のアプリケーションプログラムを起動させることによって上記の業務処理部、DB 制御部及び複数の DB 連絡部が構成され、同コンピュータ上で DB サーバ用プログラムを複数起動させることによって上記の各 DB サーバが構成され、上記 AP サーバ及び各 DB サーバには、それぞれ複数の異なる CPU コアが専用的に割り当てられており、上記 DB サーバにデータの抽出及び登録を依頼するに際しては、上記 DB 連絡部から当該処理を担当する CPU コアを具体的に指定した形で SQL 文が発行され、これを受けた DB サーバでは、指定された CPU コアによって対応の処理が実行されることを特徴としている。

[0030] 請求項 1 3 に記載したサービス提供システムの機能拡張方法は、請求項 9 のシステムにおいて、クライアント端末に対して提供するサービスの機能を拡張するに際し、拡張後の機能に対応したテーブルを上記の各 DB サーバ内に追加すると共に、拡張前の機能に係るテーブルを各 DB サーバ内に置いたまま処理の対象外に設定し、上記 AP サーバの業務処理部、DB 制御部及び各 DB 連絡

部を、追加したテーブルに対応した拡張バージョンに置き換えることを特徴としている。

発明の効果

[0031] 請求項 1 に記載したデータ管理システムによれば、距離的に離れた複数のデータセンター内に、それぞれ AP サーバと複数の DB サーバが設置され、AP サーバによって生成された業務データが同一データセンター内に設置された複数の DB サーバのみならず、他のデータセンター内に設置された複数の DB サーバにも同時に送信されるため、業務データの保全性を高めることができる。

[0032] しかも、業務データについては障害発生時にデータの復旧に複雑な処理を要する削除や更新が禁止されており、単純に他の DB サーバから不足データをコピーするだけでデータの復旧が可能な追加登録のみが許容されているため、2 相コミットのような厳格な調停方式を採用する必要がなく、各 DB サーバから単純に受取通知が返ってきた時点で登録完了と認定することができる。この結果、AP サーバと DB サーバ間における電文数を大幅に低減することが可能となり、データバックアップの迅速性を確保することができる。

[0033] 各データセンターは、複数の DB サーバのみならず、AP サーバも設置されており、平常時においても各データセンターにおいて業務処理が実行されるため、業務処理の負荷分散に貢献することができる。

各 AP サーバにおいて生成される業務データには、それぞれの AP サーバの識別符号を含む人工キーが ID として割り振られているため、各データセンター内に設置された全 DB サーバに各 AP サーバからバラバラに業務データが送信されても、ID が相互に重複することがなく、DB サーバ上でエラーとなる危険性はない。

[0034] 請求項 2 に記載したデータ管理システムの場合、業務データが複数の実体データに細分化されているため、イーサネット（登録商標）のフレームサイズ内に収容し易くなり、AP サーバと DB サーバ間の伝送効率を向上させることができる。

各実体データは、共通に保持している統合データの ID によって関連付けら

れているため、業務処理部は細分化された実体データを一連の業務データとして取り扱うことができる。

[0035] 請求項3に記載したデータ管理システムの場合、APサーバと各DBサーバ間が複数のコネクションで接続されると共に、各コネクションを介して業務データが同時並列的に送信されるため、APサーバとDBサーバ間の伝送効率をさらに向上させることができる。

[0036] この発明の場合、DBサーバのテーブルにはレコードの削除及び更新が禁止される制約が設けられているが、請求項4及び5に記載の仕組みを用いることにより、業務データの実質的な削除や更新を実現することができる。

[0037] 請求項6に記載したデータ管理システムの場合、コンピュータ上に起動されたAPサーバ及び複数のDBサーバに対して、それぞれ別個のCPUコアが専用的に割り当てられているため、同一の筐体内にAPサーバと複数のDBサーバを共存させても、CPUリソースの競合が生じることはない。

また、各DBサーバは相互に独立したテーブルを備えており、各テーブルには共通のデータが格納されているため、APサーバは複数のDBサーバに対して同時に異なるデータの抽出を依頼することで、全体における処理速度を向上させることができる。

しかも、APサーバからDBサーバにデータ抽出のSQLを発行するに際しては、当該処理を担当すべきCPUコアを指定した形でなされるため、マルチコア化されたCPUの潜在能力を極限まで引き出すことが可能となる。すなわち、業務処理に必要なデータの種類や演算の手順を知悉しているのはAPサーバであるため、同時並列的に処理可能な場面では異なるCPUコアを指定して異なるデータの抽出を依頼することで、CPUコアの稼働率を上げることができる。

APサーバは、DBサーバから抽出したデータに対し演算処理する機能を備えているため、複数のDBサーバからそれぞれ別個にデータを抽出した場合であっても、各データを結合したり、ソートやマッチングしたりする処理を、APサーバ側で有効に実行することができる。

[0038] 請求項7に記載したデータ管理システムの場合、APサーバ及び各DBサーバ

の利用可能なメモリがローカルメモリに限定され、リモートメモリへのアクセスが禁じられるため、さらなる処理の高速化が実現できる。

[0039] 請求項 8 に記載したデータ管理システムの場合、DBサーバ側の処理内容が単純化される結果、各DBサーバからの応答性能をさらに向上させることが可能となる。

[0040] 請求項 9 に記載したサービス提供システムの場合、各DBサーバに格納されたテーブルにレコードの削除及び更新が禁止される制約が課されており、APサーバから発行されるSQL文はデータの抽出及び登録を求めるものに限定されると共に、抽出したデータの演算処理はAPサーバ側で実行され、DBサーバに対して演算処理を指令する必要もないため、業務処理部やDB制御部を実現するためのプログラム中にSQL文を記載する必要が一切ない。この結果、これらプログラムの開発効率が向上し、サービスの早期リリースや機能の頻繁な拡張が可能となる。

また、DB連絡部を実現するためのプログラム中にはSQL文を記述する必要があるとしても、自己が担当するDBサーバに対して単純にデータの抽出や登録を求める内容に限定されるため、サービスの機能拡張の度に更新する必要はない。

[0041] この発明の場合、DBサーバのテーブルにはレコードの削除及び更新が禁止される制約が設けられているが、請求項 10 及び 11 に記載の仕組みを用いることにより、データの実質的な削除や更新を実現することができる。

[0042] 請求項 12 に記載したサービス提供システムの場合、同一コンピュータ上にAPサーバと複数のDBサーバが起動され、APサーバの各DB連絡部からDBサーバにデータ抽出用のSQL文を発行するに際しては処理を担当すべきCPUコアを指定した形でなされるため、マルチコア化されたCPUの潜在能力を極限まで引き出し、迅速な処理が可能となる。

[0043] 請求項 13 に記載したサービス提供システムの機能拡張方法にあつては、機能拡張に際し、各DBサーバに拡張後の機能に対応したテーブルが新設され、当該新規テーブルに基づいたサービスの提供に移行するものであり、既存

のテーブルについてはカラムが追加されることもなく、そのまま維持されることになるため、何らかの事情によって旧来のサービスに戻す必要が生じた場合にも迅速に対応可能となる。

発明を実施するための最良の形態

[0044] 図1は、この発明に係るデータ管理システム110の全体構成図である。

まず、東京に配置された第1のデータセンター112内には、第1のAPサーバ114と、第1のDBサーバ116と、第2のDBサーバ118が設置されている。

また、大阪に配置された第2のデータセンター120内には、第2のAPサーバ122と、第3のDBサーバ124と、第4のDBサーバ126が設置されている。

[0045] 第1のAPサーバ114及び第2のAPサーバ122は、それぞれ同一のアプリケーションプログラムを備えており、インターネット等の通信ネットワーク127経由でクライアント端末128から送信されたリクエストに対し、同一の業務処理を実行する機能を有する。

[0046] なお、APサーバの数に限定はなく、同一機能を備えた複数のAPサーバを各データセンター内に設置し、図示しないロードバランサを介して負荷分散させることもできる。

このように、同一データセンター内に複数のAPサーバを設けておけば、何れかのAPサーバの計画停止時あるいは計画外停止時においても、他のデータセンター内のAPサーバに切り替えることなく同一データセンター内の残りのAPサーバで代替可能となり、ダウンタイムを最小化することができる。

[0047] 第1のDBサーバ116、第2のDBサーバ118、第3のDBサーバ124及び第4のDBサーバ126は、それぞれ共通のテーブルを備えている。

DBサーバの数にも限定はなく、3台以上のDBサーバを各データセンター内に設置することもできる。

[0048] 第1のAPサーバ114は、LANを介して第1のDBサーバ116及び第2のDBサーバ118と接続されると共に、通信回線129を介して第2のデータセンター120内に設置された第3のDBサーバ124及び第4のDBサーバ126とも接続されている。

また、第2のAPサーバ122は、LANを介して第3のDBサーバ124及び第4のDB

サーバ126と接続されると共に、通信回線129を介して第1のデータセンター112内に設置された第1のDBサーバ116及び第2のDBサーバ118とも接続されている。

[0049] 平常時においては、東日本に設置されたクライアント端末128は第1のデータセンター112内に設置された第1のAPサーバ114に接続され、西日本に設置されたクライアント端末128は第2のデータセンター120内に設置された第2のAPサーバ122に接続される。

[0050] これに対し、例えば大規模な地震が首都圏で発生し、第1のデータセンター112が壊滅的な打撃を受けた場合には、DNSサーバ（図示省略）の設定変更により、東日本に設置されたクライアント端末128の接続先が第2のデータセンター120内に設置された第2のAPサーバ122に切り替えられるため、東日本のユーザに対するサービスの継続性が確保される。

同様に、第2のデータセンター120が被害を受けた場合には、西日本に設置されたクライアント端末128の接続先が第1のデータセンター112内に設定された第1のAPサーバ114に切り替えられる。

[0051] このように、各データセンターは平常時には近隣のクライアント端末128に対してサービスを提供する一方、災害時には全国のクライアント端末128に対してもサービスを提供することになるため、各DBサーバは普段から同一内容のデータを相互に保持しておく必要がある。

[0052] 図2は、第1のAPサーバ114及び第2のAPサーバ122の内部構成を示すものであり、それぞれ業務処理部130と、人工キー管理部132と、データ制御部134と、複数のDB連絡部138とを備えている。

[0053] 業務処理部130は、クライアント端末128からのリクエストに応じてデータの登録処理や検索処理等を実行するものであり、APサーバのCPUが専用のアプリケーションプログラムに従って動作することにより、実現される。

[0054] 人工キー管理部132は、業務処理部130からのリクエストに応じて、人工キー（サロゲートキー）を発行する機能を備えている（詳細は後述）。この人工キー管理部132は、専用のミドルウェアに従ってAPサーバのCPUが動作する

ことにより、実現される。

[0055] データ制御部134は、業務処理部130からデータ登録の指令を受けた際に、データをDBサーバの数だけ複製し、それぞれをDB連絡部138に渡す機能等を担うものであり、専用のミドルウェアに従ってAPサーバのCPUが動作することにより、実現される。

またデータ制御部134は、人工キー管理部132の指令を受け、同一データセンター内に設置された各DBサーバを担当するDB連絡部138に対して、人工キー管理データの更新を指令する機能をも備えている（詳細は後述）。

[0056] 各DB連絡部138は、予め自己に割り当てられたDBサーバに対してSQL文を発行し、データの登録及び抽出を指令する機能等を果たすものであり、専用のミドルウェアに従ってAPサーバのCPUが動作することにより、実現される。

[0057] このシステム110において、業務処理部130によって生成されるデータは、以下の特徴を備えている。

(1) 各データは、ユニークな人工キーからなるIDを備えている。

(2) 業務処理の結果として生じた、業務上の意味を有する実体的なデータの項目数は、一つに限定される。したがって、業務処理の結果として相互に関連する複数項目のデータが発生した場合、それぞれ独立した複数件のデータとして細分化されることになる。

(3) 相互に関連する複数のデータには、各データを束ねる統合データ（親データ）のID（統合ID）を外部キーとして共通に持たせることにより、相互の関連性が表現される。

[0058] 図3に基づいて、このデータの特徴について具体的に説明する。

例えば、クライアント端末128から予約のリクエストを受け付け、業務処理部130が対応の予約データを生成する場合、従来のシステムであれば、図3(b)に示すように、予約ID、店ID、顧客ID、金額、年月日の項目を備えた予約データ140が生成される。これらの店ID、顧客ID、金額、年月日の各項目には、1件の具体的な予約リクエストに対応する値が充填されており、相互に関連性を有している。

- [0059] これに対し、このシステム110においては、同図(a)に示すように、1件の予約リクエストに対して、予約統合データ141、予約店データ142、予約顧客データ143、予約金額データ144、予約年月日データ145の5つのデータが、業務処理部130によって生成される。
- [0060] 「予約統合データ141」は、予約店データ142、予約顧客データ143、予約金額データ144、予約年月日データ145を束ねるための上位概念的な「親データ」であり、「予約統合ID」というユニークな人工キーが充填されるデータ項目のみをプライマリーキー（PK）として備えており、業務処理的に意味のある実体的なデータ項目は備えていない。
- [0061] これに対し「予約店データ142」は、「予約店ID」というユニークな人工キーがプライマリーキー（PK）として充填されるデータ項目の他に、予約に係る店舗の特定情報が格納される「店ID」という実体的なデータ項目を備えている。また、「予約店データ142」には、自己の親データとなる予約統合IDを格納する「予約統合ID」のデータ項目が、外部キー（FK1）として設けられている。
- [0062] 同様に「予約顧客データ143」も、「予約顧客ID」というユニークな人工キーがプライマリーキー（PK）として充填されるデータ項目の他に、予約した顧客の特定情報が格納される「顧客ID」という実体的なデータ項目を備えている。また、「予約顧客データ143」にも、自己の親データとなる予約統合IDを格納する「予約統合ID」のデータ項目が、外部キー（FK1）として設けられている。
- [0063] 「予約金額データ144」も、「予約金額ID」というユニークな人工キーがプライマリーキー（PK）として充填されるデータ項目の他に、予約したサービスの金額が格納される「金額」という実体的なデータ項目を備えている。また、自己の親データとなる予約統合IDを格納する「予約統合ID」のデータ項目が、外部キー（FK1）として設けられている。
- [0064] 「予約年月日データ145」も、「予約年月日ID」というユニークな人工キーがプライマリーキー（PK）として充填されるデータ項目の他に、予約した日

にちが格納される「年月日」という実体的なデータ項目を備えている。また、自己の親データとなる予約統合IDを格納する「予約統合ID」のデータ項目が、外部キー（FK1）として設けられている。

[0065] 以上のように、予約店データ142、予約顧客データ143、予約金額データ144、予約年月日データ145は、それぞれ「予約統合ID」のデータ項目を外部キーとして有しており、それぞれに共通する予約統合IDが充填されるため、業務処理部130は各データの予約統合IDを参照することにより、細分化されたデータを統合し、相互に関連するデータからなる1件の予約データ（図3(b)参照）として取り扱うことが可能となる。

[0066] なお、予約統合データ141、予約店データ142、予約顧客データ143、予約金額データ144、予約年月日データ145は、第1のDBサーバ116、第2のDBサーバ118、第3のDBサーバ124及び第4のDBサーバ126に共通して設けられた予約統合テーブル150、予約店テーブル151、予約顧客テーブル152、予約金額テーブル153、予約年月日テーブル154に格納される。

[0067] ここで、図4のフローチャートに基づき、このシステム110における人工キー付与の仕組みについて説明する。

まず、業務処理の結果としてデータを生成する必要が生じた業務処理部130は、人工キー管理部132に対して、テーブルを指定して人工キーの発行を要求する（S110）。上記の予約データを生成するケースであれば、「予約統合ID」、「予約店ID」、「予約顧客ID」、「予約金額ID」、「予約年月日ID」用の人工キーの発行を要求することとなる。

[0068] これを受けた人工キー管理部132は、同じデータセンター内に設置されたDBサーバに格納されている人工キー管理テーブル156を参照し（S112）、それぞれの最新の人工キーを業務処理部130に通知する（S114）。

[0069] 図5に示すように、人工キー管理テーブル156は、「テーブルID」、「APサーバID」、「次のキー値」のデータ項目を備えており、「テーブルID」+「APサーバ」単位で最新の人工キー（次のキー値）が管理されている。

[0070] ここで「人工キー」とは、所定の長さ（例えば32bitまたは64bit）を備え

た数値よりなる。

また、数値の下一桁（末尾）には、IDを発行したAPサーバを特定するための識別符号（0～9の何れかの数値）がセットされている。

具体的には、図5に示すように、人工キーの下一桁管理テーブル157において、各APサーバのIDと0～9の数値との関連性が定義されている。

また、APサーバ管理テーブル158及びデータセンター管理テーブル159において、各APサーバとデータセンターとの関連性が定義されている。

[0071] 業務処理部130に対して人工キーの最新値を通知し終えた人工キー管理部132は、人工キー管理テーブル156の各レコードの値を更新させる（S116）。

具体的には、当該APサーバに係る予約統合テーブル150、予約店テーブル151、予約顧客テーブル152、予約金額テーブル153、予約年月日テーブル154の「次のキー値」に対して、「直前の値（上記の最新値）の中で、末尾の識別符号を除いた部分で構成される数値に、1を加算した値」をセットする。

[0072] ただし、人工キーの最新値の更新方法としては、このような「昇順」に限定されるものではなく、「直前の値（上記の最新値）の中で、末尾の識別符号を除いた部分で構成される数値から1を減算した値」をセットする「降順」であってもよい。APサーバの識別符号も上記のような「0～9」の数値に限定されるものではなく、他の文字種を用いてもよい。また、人工キーの先頭に識別符号を挿入してもよい。

さらには、人工キー管理テーブル156の「次のキー値」にAPサーバの識別符号を除いた数値のみを格納しておき、人工キーの発行時点で人工キー管理部132が対応の識別符号を上記数値の末尾等に付加して業務処理部130に発行することもできる。

[0073] 人工キー管理部132から最新の人工キーを受け取った業務処理部130は、当該人工キーをIDとしてプライマリーキーや外部キーにセットしたデータを生成する（S118）。

このデータは、データ制御部134及びDB連絡部138を介して全DBサーバに送信され、それぞれに設けられた対応のテーブルにインサートされる。

[0074] 上記のように、人工キーは「APサーバ×テーブル」単位でユニーク性が担保され、各APサーバはデータセンター管理テーブル159において特定のデータセンターと紐付けされているため、各APサーバにおいてそれぞれ独自に人工キーが発行されたとしても、下一桁の数値に独自性が確保されているため、重複する人工キーが発行される危険性はなく、各データのユニーク性は確実に担保される。

[0075] なお、APサーバの業務処理部130から発せられた人工キー発行のリクエストは、一旦キューに溜められた上で、FIFO（先入れ先出し）のルールに則って処理されるため、後から発生したデータに対して誤って値の小さい人工キーが発行される危険性はなく、順序性が担保される。

[0076] 各APサーバの業務処理部130によって生成されたデータは、上記のように各DBサーバ内のテーブルに格納されることになるが、各テーブルには、デリート（削除）禁止及びアップデート（更新）禁止の制約が予め課せられている。

要するに、このシステム110においては、各テーブルに対するインサート（挿入）とセレクト（参照）のみが許容されることになる。

[0077] このため、既存のデータを無効にする必要が生じた場合には、レコードを削除する代わりに、他のテーブルに新たなレコードをインサートすることで同等の状態を実現する。

[0078] 例えば、図3に示した1件の予約データ（予約統合データ＋予約店データ＋予約顧客データ＋予約金額データ＋予約年月日データ）を無効化する場合には、図6(a)に示すように、取消統合データ160＋取消対象データ161＋取消年月日データ162からなる予約取消用の各データを、それぞれ取消統合テーブル163、取消対象テーブル164、取消年月日テーブル165にインサートする。

[0079] 「取消統合データ160」は、取消対象データ161、取消年月日データ162を束ねるための上位概念的な「親データ」であり、「取消統合ID」というユニークな人工キーが充填されるデータ項目のみをプライマリーキー（PK）として備えており、業務処理的に意味のある実体的なデータ項目は備えていない。

- [0080] これに対し「取消対象データ161」は、「取消対象ID」というユニークな人工キーがプライマリーキー（PK）として充填されるデータ項目の他に、取消対象となる予約データの予約統合IDが格納される「取消対象」という実体的なデータ項目を備えている。また、自己の親データとなる取消統合IDを格納する「取消統合ID」のデータ項目が、外部キー（FK1）として設けられている。
- [0081] 「取消年月日データ162」も、「取消年月日ID」というユニークな人工キーがプライマリーキー（PK）として充填されるデータ項目の他に、取消年月日が格納される「年月日」という実体的なデータ項目を備えている。また、自己の親データとなる取消統合IDを格納する「取消統合ID」のデータ項目が、外部キー（FK1）として設けられている。
- [0082] 以上のように、取消対象データ、取消年月日データは、それぞれ「取消統合ID」のデータ項目を外部キーとして有しており、それぞれに共通する取消統合IDが充填されるため、業務処理部130は各データの取消統合IDを参照することにより、細分化されたデータを統合し、相互に関連するデータからなる1件の予約取消データ166（図6(b)参照）として取り扱うことが可能となる。
- [0083] そして、予約統合テーブル150、予約店テーブル151、予約顧客テーブル152、予約金額テーブル153、予約年月日テーブル154から予約データをセレクトした後に、取消統合テーブル163、取消対象テーブル164、取消年月日テーブル165を参照し、当該予約データの「統合取消ID」が取消対象データ161の「取消対象」に格納されている場合には、当該予約データを無効なものとして除外する。
- [0084] なお、取消統合データ160、取消対象データ161、取消年月日データ162は、第1のDBサーバ116、第2のDBサーバ118、第3のDBサーバ124及び第4のDBサーバ126に共通して設けられた取消統合テーブル163、取消対象テーブル164、取消年月日テーブル165にそれぞれ格納される。
- [0085] また、このシステム110において登録データの修正が必要な場合には、既存

レコードをアップデートする代わりに、修正後のデータを備えたレコードを新規に登録することで対応する。

例えば、図7に示すように、予約店テーブルに格納された「予約店ID：12345678901」＋「予約統合ID：22345678901」＋「店ID：M012356」のレコード167に関して、店IDを「店ID：M012357」に変更する必要がある場合、業務処理部130は既存レコード167の値を変更する代わりに、修正後の店IDを含む新たなレコード168をインサートする。

[0086] この新しいレコード168は、予約統合IDとして既存のレコード167と等しい値が設定されている一方、既存の予約店IDよりも値の大きな予約店ID（22345678901）が割り振られているため、業務処理部130は新規レコード168が修正後の値を表していることを認識でき、レコードのインサートによって実質的にデータの更新を実現することが可能となる。

なお、図示の便宜上、図7においては「予約店ID：12345678901」、「予約統合ID：22345678901」、「予約店ID：24567890141」として実際に用いられる人工キーよりも短いものが例示されている。

[0087] 業務処理部130によって生成されたデータは、データ制御部134によってDBサーバの数だけ複製された後、各DBサーバを専門に担当するDB連絡部138を介して全DBサーバに対し一斉に送信される。

図8においては、「(1)予約統合データ141」、「(2)予約店データ142」、「(3)予約顧客データ143」、「(4)予約金額データ144」、「(5)予約年月日データ145」が、第1のDBサーバ116～第4のDBサーバ126に送信される様子が示されている。

[0088] この際、各データのDBサーバへの到達順序が問われることはなく、親データである「(1)予約統合データ141」よりも先に子データである「(5)予約年月日データ145」等がDBサーバに到達していてもエラーになることはない。

これは、各子データには「(1)予約統合データ141」の「予約統合ID」が外部キーとして割り振られているため、到達順序が先後しても業務処理部130は最終的に各データを正しく束ねることが可能だからである。

[0089] 各DBサーバは、DB連絡部138からデータを受け取って自身のメモリに格納した時点で、DB連絡部138に対して受取完了通知を送信し、その後、ハードディスクやSSD等の外部記憶装置内に設けられたテーブルに対応のレコードを格納する。

[0090] このDBサーバから送信された受取完了通知は、各DB連絡部138を経由してデータ制御部134に集められる。

データ制御部134は、全てのDB連絡部138から「(1)予約統合データ141」、「(2)予約店データ142」、「(3)予約顧客データ143」、「(4)予約金額データ144」、「(5)予約年月日データ145」の全データについて受取完了通知が返ってきた時点で、業務処理部130に対して登録完了通知を出力する。

[0091] 業務処理部130は、この登録完了通知をデータ制御部134から受け取った時点で、上記データを読み込みの対象と認識する。逆に言えば、登録完了通知が返ってくるまでの間、業務処理部130は当該データを読み込みの対象外とすることで、各DBサーバにおける登録のタイミングがずれて不正なデータが読み込まれてしまう危険性を回避している。

[0092] ただし、何れかのDBサーバから一定時間を経過しても受取完了通知が返って来ない場合、データ制御部134は通信障害あるいはマシントラブルが発生したものと認定し、当該DBサーバをオフラインモードに移行させる。

具体的には、当該DBサーバを一時的にシステム110から切り離し、残りのDBサーバのみに基づく運用形態に移行する。

すなわち、切り離されたDBサーバ以外の全DBサーバから受取完了通知が届いていた場合、データ制御部134から業務処理部130に対して登録完了通知が出力され、登録データが読み取りの対象となる。

[0093] この間、切り離されたDBサーバについて点検がなされ、必要な復旧対策が施される。例えば、通信機器の故障が原因で受取完了通知が届いていなかったと判明した場合、新しい通信機器に交換した上で、当該DBサーバをライトモード（データの書込のみが許容され、データの参照が禁止される状態）に設定し、同一データセンター内に設置された他のDBサーバから差分データを

コピーする。

[0094] そして、格納データが最新の状態に迫り着いた時点で、当該DBサーバはリード／ライトモード（データの書込及び参照が許容される状態）に設定され、システム110に再接続される。

以後、データ制御部134は、当該DBサーバを担当しているDB連絡部138を介して、データの送受信を再開する。

[0095] このシステム110の場合、各業務データはインサートのみが許容され、デリートやアップデートが認められないルールが適用されているため、データの復旧が極めて容易となる。

すなわち、データの削除や更新が許容されることを前提とした場合、データを復旧させるためにはDBサーバが保持している更新履歴情報に基づき、一定の時点からデータの追加や削除、更新を順番に再現する必要がある、これに長い時間と大きな負荷を要することになる。また、更新ログを確実に保存する仕組みをDBサーバ側に設ける必要がある。

[0096] これに対し、このシステム110のようにデータのインサートのみが許容され、削除と更新が禁止される前提の場合、他の正常なDBサーバに登録されたデータと障害が発生したDBサーバに登録されたデータを比較し、単純に足りないデータをコピーするだけで済み、データの順序（先後関係）を考慮する必要がなくなる。

[0097] また、このシステム110にあっては、上記のようにDB連絡部138から送信されたデータがDBサーバのメモリに格納された時点で、DBサーバから受取完了通知がAPサーバに発行されるため、極めて短時間の中に業務処理部130は当該データを読み込みの対象とすることができる。

[0098] これまでの常識では、メモリは揮発性の記憶手段であり、電源供給が停止された時点でデータが喪失してしまうため、不揮発性の記憶手段（ハードディスク等）にデータが格納された時点で完了通知が返されるべきとなる。

これに対し、このシステム110の場合には、あるDBサーバでデータの欠損が発生しても、単純に隣接するDBサーバから不足データをコピーするだけで追

い着くことができるため、データがハードディスク等に格納されるまで待機する必要がない。

[0099] また、このようにデータの復旧が容易であるため、複数のDBサーバに同一データを格納するに際し、従来のように2相コミットという面倒な手順を踏むことが求められず、各DBサーバに対して単純に登録を依頼し、それぞれから受取完了通知が返ってきた時点でリード可能とすることが可能となる（いわゆるオートコミットの実現）。

この結果、APサーバとDBサーバ間の電文のやり取りが激減し、全体的な処理速度を向上させることができる。

[0100] また、このシステム110の場合、上記のように業務処理部130によって生成される各データが細分化され、1件の送信データの容量が極小化されているため、イーサネット（登録商標）のフレームサイズ内に多くのデータを収容し易くなり、APサーバとDBサーバ間の伝送効率を向上させることができる。

[0101] さらに、図9に示すように、APサーバと各DBサーバとの間にそれぞれ複数のコネクション171～175を形成し、それぞれのコネクションと専用のDB連絡部138を介して細分化されたデータを同時並列的に送信することも可能となり、APサーバとDBサーバ間の伝送効率をさらに高めることが可能となる。

[0102] ただし、このように業務処理部130によって生成されるデータがデータ項目単位に細分化され、多数のDB連絡部138を用いた並列処理が実行されるとなると、APサーバ側の処理が複雑化することは否めない。これについては、APサーバ側をマルチコア化することで対応可能となる。

[0103] すなわち、APサーバとして多数のCPUコアを搭載したコンピュータを採用すると共に、OSやミドルウェアの機能によって多数のスレッドを起動させ、各スレッドによって業務処理部130や人工キー管理部132、多数のDB連絡部138を実現するマルチタスク方式を採用することで、APサーバ側の処理を効率化することができる。

[0104] 図10は、CPUコア180とスレッド181との対応関係を表しており、各スレッド181にはスレッドプール182が設けられ、そこに配置された複数のタスク183

をスレッド181が順次処理していくイメージが描かれている。

この図におけるスレッド181が、各APサーバの業務処理部130、人工キー管理部132、データ制御部134、DB連絡部138として機能し、これらの各機能構成部が実行する具体的な処理が、タスク183に相当する。

[0105] 各スレッド181は、スレッドプール182に蓄積されたタスク183を古い順に次々と実行していき、自己のスレッドプール182が空になった場合には、他のスレッド181のスレッドプール182に蓄積されたタスク183を、新しい順に実行していく。

[0106] なお、この発明にとって業務データの細分化は必須の構成要件ではなく、相互に関連する複数の実体データを1件のデータとしてまとめ、これに人工キーよりなるIDを付加した業務データ（図3(b)及び図6(b)参照）を業務処理部130が生成し、これをデータ制御部134及び複数のDB連絡部138を介して各DBサーバに送信するように構成することもできる。

[0107] 上記の各種業務データ（予約統合データ141、予約店データ142、予約顧客データ143、予約金額データ144、予約年月日データ145等）には、業務処理部130によってミリ秒精度のタイムスタンプが刻印されているため、人工キーの値がMAXに達し、発行済のIDと重複するIDの再発行（リサイクル）がなされた場合でも、業務処理部130はタイムスタンプを比較することで新旧データの判定が可能となる。

[0108] 上記においては、APサーバとDBサーバが物理的に独立して存在していることを前提に説明を展開したが、同一データセンター内であれば、APサーバとDBサーバを同一筐体のサーバコンピュータによって構成することもできる。

[0109] 図11は、この発明に係る他のデータ管理システム210を実現するためのハードウェア構成を示すものであり、サーバコンピュータ211内には、第1のCPUパッケージ212、第2のCPUパッケージ214、第3のCPUパッケージ216、第4のCPUパッケージ218を備えたマザーボード220が収納されている。

[0110] 各CPUパッケージ内には、それぞれ15個のCPUコアが形成されているため、サーバコンピュータ211は合計で60個ものCPUコアを有していることとなる。

各CPUパッケージは、マザーボード220上に設けられたCPUソケット（図示省略）にそれぞれ装着されている。

[0111] 各CPUパッケージには、それぞれ専用のメモリ（ローカルメモリ）が割り当てられている。

すなわち、第1のCPUパッケージ212には第1のメモリ222が、第2のCPUパッケージ214には第2のメモリ224が、第3のCPUパッケージ216には第3のメモリ226が、第4のCPUパッケージ218には第4のメモリ228が専属的に設けられている。

[0112] サーバコンピュータ211には、SSD（Solid State Drive）よりなる外部記憶装置230が接続されており、この外部記憶装置230内には各種テーブル（詳細は後述）が格納されている。SSDの代わりに、ハードディスクによって外部記憶装置230を構成することもできる。

[0113] 各CPUコアには、それぞれ固有の機能が予め割り当てられている。

すなわち、第1のCPUパッケージ212内に形成されたコア(01)～コア(15)には、APサーバ232が専用的に割り当てられている。

[0114] また、第2のCPUパッケージ214内に形成されたコア(16)には、カーネルスレッド（OS）234が専用的に割り当てられている。

第2のCPUパッケージ214内に形成されたコア(17)～コア(23)には、第1のDBサーバ236が専用的に割り当てられている。

同じく、第2のCPUパッケージ214内に形成されたコア(24)～コア(30)には、第2のDBサーバ238が専用的に割り当てられている。

[0115] 第3のCPUパッケージ216内に形成されたコア(31)～コア(38)には、第3のDBサーバ240が専用的に割り当てられている。

同じく、第3のCPUパッケージ216内に形成されたコア(39)～コア(45)には、第4のDBサーバ242が専用的に割り当てられている。

[0116] 第4のCPUパッケージ218内に形成されたコア(46)～コア(53)には、第5のDBサーバ244が専用的に割り当てられている。

同じく、第4のCPUパッケージ218内に形成されたコア(54)～コア(60)には

、第6のDBサーバ246が専用的に割り当てられている。

[0117] 以上のように、1台のサーバコンピュータ211内には、1つのAPサーバ用プログラムと、6つのDBサーバ用プログラム（インスタンス）が起動されており、各サーバにはそれぞれ複数のCPUコアが専用的に割り当てられている。

[0118] また、上記した通り、APサーバ232には第1のメモリ222が、カーネルスレッド（OS）234、第1のDBサーバ236及び第2のDBサーバ238には第2のメモリ224が、第3のDBサーバ240及び第4のDBサーバ242には第3のメモリ226が、第5のDBサーバ244及び第6のDBサーバ246には第4のメモリ228が、それぞれローカルメモリとして専用的に割り当てられている。

この場合、各CPUコアはローカルメモリにはダイレクトに（高速に）アクセスできるが、他のCPUパッケージに割り当てられたリモートメモリにアクセスするためには、当該CPUパッケージに属するCPUコアを経由する必要がある、その分、処理速度が低下する問題が生じる。

このため、このシステム210においては、各CPUコアがアクセスできるのはローカルメモリのみに限定され、他のCPUパッケージの配下にあるリモートメモリへのアクセスは禁止されている。

[0119] このような割当て及び制約は、OS起動時に次のような「NUMAコントロール」のコマンドを入力することにより、実現される。

[0120] [1] APサーバに関する割当て

```
#numactl
    -physcpubind=1-15    ←コア(01)～(15)の割当て
    -membind=0           ←第1のメモリの割当て
    nice -n -5           ←プロセスの優先度の指定
    java -jar aloha.jar   ←APサーバプロセス（Java）の起動
```

[0121] [2] 第1のDBサーバに関する割当て

```
#numactl
    -physcpubind=17-23   ←コア(17)～(23)の割当て
    -membind=1           ←第2のメモリの割当て
```

```
nice -n -5                ←プロセスの優先度の指定
su -postgres              ←プロセスの実行ユーザをpostgresに切り替え
(su)
-c "pg_ctl -D data1 start" ←第1のDBサーバプロセスの起動
(データ領域data1に対するデータベース) 」
```

[0122] [3] 第2のDBサーバに関する割当て

```
#numactl
-physcpubind=24-30        ←コア(24)～(30)の割当て
-membind=1                ←第2のメモリの割当て
nice -n -5                ←プロセスの優先度の指定
su -postgres              ←プロセスの実行ユーザをpostgresに切り替
え (su)
-c "pg_ctl -D data2 start" ←第2のDBサーバプロセスの起動
(データ領域data2に対するデータベース) 」
```

[0123] [4] 第3のDBサーバに関する割当て

```
#numactl
-physcpubind=31-38        ←コア(31)～(38)の割当て
-membind=2                ←第3のメモリの割当て
nice -n -5                ←プロセスの優先度の指定
su -postgres              ←プロセスの実行ユーザをpostgresに切り替え
(su)
-c "pg_ctl -D data3start" ←第3のDBサーバプロセスの起動
(データ領域data3に対するデータベース) 」
```

[0124] [5] 第4のDBサーバに関する割当て

```
#numactl
-physcpubind=39-45        ←コア(39)～(45)の割当て
-membind=2                ←第3のメモリの割当て
nice -n -5                ←プロセスの優先度の指定
```

```
su -postgres          ←プロセスの実行ユーザをpostgresに切り替え  
(su)
```

```
-c "pg_ctl -D data4 start" ←第4 DBサーバプロセスの起動  
(データ領域data4に対するデータベース) 」
```

[0125] [6] 第5のDBサーバに関する割当て

```
「#numactl  
-physcpubind=46-53    ←コア(46)～(53)の割当て  
-membind=3           ←第4のメモリの割当て  
nice -n -5           ←プロセスの優先度の指定  
su -postgres          ←プロセスの実行ユーザをpostgresに切り替え  
(su)  
-c "pg_ctl -D data5 start" ←第5のDBサーバプロセスの起動  
(データ領域data5に対するデータベース) 」
```

[0126] [7] 第6のDBサーバに関する割当て

```
「#numactl  
-physcpubind=54-60    ←コア(54)～(60)の割当て  
-membind=3           ←第4のメモリの割当て  
nice -n -5           ←プロセスの優先度の指定  
su -postgres          ←プロセスの実行ユーザをpostgresに切り替え  
(su)  
-c "pg_ctl -D data6 start" ←第6のDBサーバプロセスの起動  
(データ領域data6に対するデータベース) 」
```

[0127] 図12は、このデータ管理システム210の機能構成を示すブロック図であり、APサーバ232内には、画面処理部252と、処理内容特定部254と、DB処理割当部256と、演算処理部258が形成されている。

画面処理部252には、インターネット等の通信ネットワーク260を介して、複数のクライアント端末262が接続される。

[0128] これら画面処理部252、処理内容特定部254、DB処理割当部256及び演算処理

部258は、業務処理用のアプリケーションプログラムに従い、CPUコア(01)～コア(15)が所定の処理を実行することにより実現されるのであるが、この際、OSやミドルウェアによって複数のスレッドが起動されて各CPUコアに割り当てられることにより、マルチタスク処理が実現される。

[0129] 図13は、各CPUコアと複数のスレッド264との対応関係を表しており、各スレッド264にはスレッドプール266が設けられ、そこに配置された複数のタスク268をスレッド264が順次処理していくイメージが描かれている。

この図におけるスレッド264が、APサーバ232の画面処理部252、処理内容特定部254、DB処理割当部256、演算処理部258として機能し、これらの各機能構成部が実行する具体的な処理が、タスク268に相当する。

[0130] 各スレッド264は、スレッドプール266に蓄積されたタスク268を古い順に次々と実行していき、自己のスレッドプール266が空になった場合には、他のスレッド264のスレッドプール266に蓄積されたタスク268を、新しい順に実行していく。

[0131] DB処理割当部256は、第1のDBサーバ236～第6のDBサーバ246に対してSQLを発行し、業務処理に必要なデータの抽出（セレクト）や追加（インサート）を指令する機能を果たす。

[0132] DB処理割当部256と各DBサーバとの間は、それぞれのCPUコア数に対応した本数のコネクションプール270を介して接続されている。

例えば、第1のDBサーバ236には7個のCPUコアが割り当てられているため7本のコネクションプール270が設けられ、各コネクションプール270は予め特定のCPUコアと関連付けられている。

また、第5のDBサーバ244には8個のCPUコアが割り当てられているため8本のコネクションプール270が設けられ、各コネクションプール270は予め特定のCPUコアと関連付けられている。

[0133] 外部記憶装置230内には、例えば、仕入テーブル272、注文テーブル273、注文明細テーブル274、注文取消テーブル275、品目テーブル276、完成品テーブル277、原材料テーブル278、原材料目録テーブル279、原価テーブル280、供

給者テーブル281等を含むテーブル群が、各DBサーバ単位で複数セット格納されている。

ここでは、第1のDBサーバ236～第6のDBサーバ246が設けられているため、A～Fの6セットのテーブル群が、それぞれ各DBサーバ専用として外部記憶装置230内の別個の領域に格納されている。

また、後述のように、各テーブル群に属する同一名称のテーブルには、それぞれ同一のレコードが同時に重複して格納される。

[0134] 図14は、各テーブルに格納されているレコードの構成を例示するものである。

すなわち、仕入テーブル272には、「仕入ID」、「原価ID」、「数量」のデータ項目を備えたレコードが格納されている。

また、注文テーブル273には、「注文ID」、「顧客ID」のデータ項目を備えたレコードが格納されている。

注文明細テーブル274には、「注文明細ID」、「注文ID」、「品目ID」、「数量」のデータ項目を備えたレコードが格納されている。

注文取消テーブル275には、「注文明細ID」のデータ項目を備えたレコードが格納されている。

[0135] 品目テーブル276には、「品目ID」のデータ項目を備えたレコードが格納されている。

完成品テーブル277には、「品目ID」のデータ項目を備えたレコードが格納されている。

原材料テーブル278には、「品目ID」のデータ項目を備えたレコードが格納されている。

原材料目録テーブル279には、「原材料目録ID」、「親品目ID」、「子品目ID」、「数量」のデータ項目を備えたレコードが格納されている。

原価テーブル280には、「原価ID」、「供給者品目ID」、「原価」のデータ項目を備えたレコードが格納されている。

供給者テーブル281には、「供給者品目ID」、「供給者ID」、「品目ID」の

データ項目を備えたレコードが格納されている。

[0136] 上記の各テーブル272～281に関しては、以下の(1)～(3)の制約が予め課せられている。

(1)キー項目は一つに限定される。

したがって、複数の項目の組合せによってキー項目を構成すること（複合キー）は、禁止される。

[0137] (2)APサーバ232が各テーブルにアクセスするに際しては、レコードの参照と追加のみが許容され、既存レコードの削除と更新は禁止される。

[0138] このため、既存データを無効にする必要が生じた場合には、レコードを削除する代わりに、他のテーブルに新たなレコードを追加することによって削除と同等の状態が実現される。例えば、注文の取り消しがなされた場合、一般的には取消の対象となる注文データに「取消」を示す値を記述することで状態が管理されることになるが、このシステム210においては、注文取消テーブル275に対象となる「注文明細ID」のレコードを追加することで「取消」の事実が表現される。

[0139] また、既存レコードの値に変更が生じた場合には、修正済みの値を備えた異なるプライマリーキーのレコードを同一テーブルに追加することで、実質的に値の更新が実現される。例えば、注文明細テーブル274に格納されたあるレコードの数量を「3」から「5」に変更する場合、数量を「5」とした新たなレコードが注文明細テーブル274に追加される。各レコードにはミリ秒精度のタイムスタンプが刻印されているため、これを比較することでレコード間の新旧（更新の事実）が明確化される。

[0140] (3)各テーブルのカラムには、NULL値禁止制約、フラグ値禁止制約、区分値禁止制約が課せられる。

まず「NULL値禁止制約」とは、データ項目の値としてNULL（値なし）を充填することが禁止されることを意味しており、このようなNULL値の充填を想定したデータ項目の設定自体が許容されないことになる。

つぎに「フラグ値禁止制約」とは、データ項目の値としてフラグ値（「1

／O」、「ON／OFF」、「TRUE／FALSE」等の2値データ)を充填することが禁止されることを意味しており、このようなフラグ値の充填を想定したデータ項目の設定自体が許容されないことになる。

「区分値禁止制約」とは、データ項目の値として区分値(「1:正社員」、「2:パート」、「3:アルバイト」等)を充填することが禁止されることを意味しており、このような区分値の充填を想定したデータ項目の設定自体が許容されないことになる。

[0141] 上記のようにAPサーバ232によるレコードの削除及び更新が禁止されているため、各テーブルには過去の如何なる時点のデータも温存されることとなり、将来において変更履歴の再現が必要となった場合にも対応できる利点がある。

ただし、十分に時間が経過し、データ保持の必要性が完全になくなった場合には、各DBサーバに直接ログインして削除のSQLを発行することにより、レコードを削除することはできる。

[0142] このシステム210の場合、各テーブルに格納される個々のレコードが上記の通り極限まで正規化されており、またデータ間の演算処理はAPサーバ232の演算処理部258によって行われ、DBサーバ側の演算処理によって冗長な中間データが生成されることもないため、レコードの削除や更新の代わりにレコードの挿入によって対応するとしても、ディスク容量が従来に比べて徒に圧迫されることはない。また、キャッシュメモリのヒット率が向上するため、DBサーバの処理の高速化にも資する。

[0143] これらの制約ルールは、新規テーブルの設計時やSQL文発行時に、各DBサーバのデータベース管理システムによって適否がチェックされ、制約ルールに違反する処理の実行が拒絶されることにより、その実効性が担保される。

[0144] このようなテーブル群によって業務データが管理されている状況下において、新規データの登録を希望するユーザは、画面処理部252からクライアント端末262に送信されたデータ入力画面(図示省略)を通じて、データを入力する。

[0145] この入力データを画面処理部252から受け取った処理内容特定部254は、DB処理割当部256に対してデータの登録を依頼する。

DB処理割当部256は、この入力データの登録を依頼するSQLを第1のDBサーバ236～第6のDBサーバ246に発行する。この際、DB処理割当部256は各DBサーバに対し、特定のCPUコアに関連付けられたコネクションプールを通じてSQLを発行することにより、各DBサーバにおける担当CPUコアを指定する。

[0146] これを受けた各DBサーバでは、DB処理割当部256によって指定されたCPUコアにより、対応のテーブルに対するレコードの挿入が一斉に実行される。

この結果、各DBサーバが独立して管理している同名テーブルに、同一レコードが重複して格納されることとなる。

[0147] つぎに、ある完成品（例えば品目ID「NRI-0001」）の在庫数についてクライアント端末から照会リクエストが送信された場合の処理手順について、図15及び図16のフローチャートに従い説明する。

[0148] まず、画面処理部252から送信された検索条件指定画面上でユーザが完成品である品目ID「NRI-0001」の在庫検索を指定すると、この検索リクエストの実現に必要な処理が処理内容特定部254によって特定される（図15のS210）。

[0149] 具体的には、以下の処理内容が特定された後、順次実行される。

(01)注文明細テーブルから、品目ID「NRI-0001」を備えた全レコードを抽出する（S212）。

(02)品目ID「NRI-0001」の累積注文数量を算出する（S214）。

(03)注文取消テーブルから、品目ID「NRI-0001」を備えた全レコードを抽出する（S216）。

(04)品目ID「NRI-0001」の累積注文取消数量を算出する（S218）。

(05)品目ID「NRI-0001」の累積注文数量から累積注文取消数量を減算して、現時点における累積実注文数量を算出する（S220）。

(06)原材料目録テーブルを参照し、品目ID「NRI-0001」を親品目IDとしている子品目IDと、その数量を取得する（S222）。この子品目IDが、完成品であ

る品目ID「NRI-0001」の原材料を示している。

(07)各子品目IDをキーにして、供給者テーブルから供給者品目IDを取得する（図16のS224）。

(08)供給者品目IDをキーにして、原価テーブルから原価IDを取得する（S226）。

(09)原価IDをキーにして、仕入テーブルから当該原材料に係る全レコードを抽出する（S228）。

(10)原材料毎に、累積仕入数量を算出する（S230）。

(11)各原材料の累積仕入数量に基づいて、品目ID「NRI-0001」の生産可能数量を算出する（S232）。

(12)この生産可能数量から累積実注文数量を減算することにより、当該完成品の現在における在庫数量を算出する（図15のS234）。

(13)最後に、「NRI-0001」の在庫数量が記載された処理結果画面をクライアント端末262に送信する（S236）。

[0150] これらの処理の中、データの抽出処理は、DB処理割当部256から各DBサーバに対してSQLを発行することによって実行される。この際、DB処理割当部256は各DBサーバに対し、特定のCPUコアに関連付けられたコネクションプールを通じてSQLを送信する。この結果、DBサーバにおけるレコードの抽出処理は、指定されたCPUコアによって実行される。

また、各DBサーバから送信されたデータに対しては、演算処理部258によってソートやマージ、マッチング、コントロールブレイク等の必要な演算処理が施される。

[0151] ところで、上記の各処理の中には、先行する処理が完了しないと後続の処理を実行できないものと、同時並行して処理が可能なものがある。

例えば、図15に示したように、注文明細テーブル274から該当品目IDのレコードを抽出するS212の処理と、注文取消テーブル275から該当品目IDのレコードを抽出するS216の処理と、原材料目録テーブル279を参照し、該当品目IDを親品目IDとしている子品目ID及び数量を取得するS222の処理は、品目

IDが特定された時点で同時並行することができる。

このため、DB処理割当部256は、上記の処理をそれぞれ別個のCPUコアを指定して何れかのDBサーバに抽出処理を依頼する。

[0152] また、S 222の処理によって完成品を生産するのに必要な原材料の種類が特定された時点で、図 1 6 に示したように、各原材料の仕入数量を算出するのに必要なデータの抽出処理が開始されることになるが、このS 224～S 228も原材料毎に同時並行で処理される。

例えば、原材料の数が250種類であった場合、DB処理割当部256は、それぞれ異なる50個のCPUコアを指定して、各DBサーバにレコードの抽出を依頼する。

このように、特定のCPUコアに具体的なレコードの抽出を直接的に割り振ることにより、効率的にデータの参照が可能となる。

[0153] なお、原材料の種類が数百種以上に及ぶ場合、原材料単位の処理を一度に異なるCPUコアに割り当てることができなくなる。

このような場合、DB処理割当部256は各CPUコアに順番に原材料毎の処理を依頼し、処理が完了した時点で当該CPUコアに残りの原材料単位の処理を順番に割り当ててを繰り返す。

[0154] 特定の業務処理を実現するために必要なデータの種類や抽出手順についてはAPサーバ232のアプリケーションプログラムが認識しているため、具体的なレコードの抽出を各DBサーバに対しCPUコアを指定した形で依頼することにより、マルチコアCPUの潜在能力を最も効果的に引き出すことが可能となる。

[0155] なお、OSには統計情報に基づいてデータ処理の効率化や省電力化を実現するため、処理の途中で敢えてCPUのクロック数をダウンさせる機能が設けられているが、本システム210のようにCPUコア単位で処理内容をきめ細かく制御する際には邪魔になるため、この種の余計なサービスは事前に無効化しておくことが望ましい。

[0156] 図 1 7 は、この発明に係るサービス提供システム310の全体構成図である。

まず、データセンター312内には、APサーバ314と、第 1 のDBサーバ316と、

第2のDBサーバ317と、第3のDBサーバ318と、第4のDBサーバ319と、第5のDBサーバ320と、第6のDBサーバ321と、第7のDBサーバ322が設置されている。

[0157] 詳細は後述するが、APサーバ314と第1のDBサーバ316～第6のDBサーバ321は同一のサーバコンピュータ324内に設けられており、マザーボード上の信号線を介して接続されている。

これに対し第7のDBサーバ322は、別筐体のサーバコンピュータ上に設けられており、LAN経由でAPサーバ314と接続されている。

[0158] またAPサーバ314は、遠隔地に設置された第8のDBサーバ326、第9のDBサーバ328及び第10のDBサーバ330と、通信回線332を介して接続されている。

さらにAPサーバ314は、インターネット334を介して、複数のクライアント端末336とも接続されている。

[0159] 第1のDBサーバ316～第10のDBサーバ330には、それぞれ共通のテーブルが設けられている（詳細は後述）。

[0160] 図18は、APサーバ314の内部構成を示すものであり、業務処理部340と、DB制御部342と、第1のDB連絡部343と、第2のDB連絡部344と、第3のDB連絡部345と、第4のDB連絡部346と、第5のDB連絡部347と、第6のDB連絡部348と、第7のDB連絡部349と、第8のDB連絡部350と、第9のDB連絡部351と、第10のDB連絡部352を備えている。

[0161] 業務処理部340は、クライアント端末336からのリクエストに応じてデータの登録や検索等の業務処理を実行するものであり、サーバコンピュータ324のCPUが専用のアプリケーションプログラムに従って動作することにより、実現される。

[0162] 第1のDB連絡部343～第10のDB連絡部352には、それぞれ第1のDBサーバ316～第10のDBサーバ330が予め専属的に割り当てられており、担当のDBサーバに対してSQL文を発行し、データの登録及び抽出を指令する機能を果たす。

各DB連絡部は、専用のミドルウェアに従ってサーバコンピュータ324のCPUが動作することにより、実現される。

[0163] DB制御部342は、業務処理部340からデータ登録の指令を受けた際に、データをDBサーバの数だけ複製し、それぞれを各DB連絡部に渡す機能と、業務処理部340からデータ参照の指令を受けた際に、何れかのDB連絡部を介して特定のDBサーバからデータを抽出し、必要な演算処理を施す機能を果たす。

このDB制御部342は、専用のアプリケーションプログラムに従ってサーバコンピュータ324のCPUが動作することにより、実現される。

[0164] 図19は、サーバコンピュータ24内部のハードウェア構成例を示すものであり、第1のCPUパッケージ360、第2のCPUパッケージ362、第3のCPUパッケージ364、第4のCPUパッケージ366を備えたマザーボード368が筐体内に収納されている。

[0165] 各CPUパッケージ内には、それぞれ15個のCPUコアが形成されているため、サーバコンピュータ324は、合計で60個ものCPUコアを有していることとなる。

各CPUパッケージは、マザーボード368上に設けられたCPUソケット（図示省略）にそれぞれ装着されている。

[0166] 各CPUパッケージには、それぞれ専用のメモリ（ローカルメモリ）が割り当てられている。

すなわち、第1のCPUパッケージ360には第1のメモリ370が、第2のCPUパッケージ362には第2のメモリ371が、第3のCPUパッケージ364には第3のメモリ372が、第4のCPUパッケージ366には第4のメモリ373が専属的に設けられている。

[0167] サーバコンピュータ324には、SSD（Solid State Drive）よりなる補助記憶装置374が接続されており、この補助記憶装置374内には各種テーブル（詳細は後述）が格納されている。SSDの代わりに、ハードディスクによって補助記憶装置374を構成することもできる。

[0168] 各CPUコアには、それぞれ固有の機能が予め割り当てられている。

すなわち、第1のCPUパッケージ360内に形成されたコア(01)～コア(15)には、APサーバ314が専用的に割り当てられている。

[0169] また、第2のCPUパッケージ362内に形成されたコア(16)には、カーネルスレッド(OS) 376が専用的に割り当てられている。

第2のCPUパッケージ362内に形成されたコア(17)～コア(23)には、第1のDBサーバ316が専用的に割り当てられている。

同じく、第2のCPUパッケージ362内に形成されたコア(24)～コア(30)には、第2のDBサーバ317が専用的に割り当てられている。

[0170] 第3のCPUパッケージ364内に形成されたコア(31)～コア(38)には、第3のDBサーバ318が専用的に割り当てられている。

同じく、第3のCPUパッケージ364内に形成されたコア(39)～コア(45)には、第4のDBサーバ319が専用的に割り当てられている。

[0171] 第4のCPUパッケージ366内に形成されたコア(46)～コア(53)には、第5のDBサーバ320が専用的に割り当てられている。

同じく、第4のCPUパッケージ366内に形成されたコア(54)～コア(60)には、第6のDBサーバ321が専用的に割り当てられている。

[0172] 以上のように、1台のサーバコンピュータ324内には、1つのAPサーバ用プログラムと、6つのDBサーバ用プログラム(インスタンス)が起動されており、各サーバにはそれぞれ複数のCPUコアが専用的に割り当てられている。

[0173] また、上記した通り、APサーバ314には第1のメモリ370が、カーネルスレッド(OS) 376、第1のDBサーバ316及び第2のDBサーバ317には第2のメモリ371が、第3のDBサーバ318及び第4のDBサーバ319には第3のメモリ372が、第5のDBサーバ320及び第6のDBサーバ321には第4のメモリ373が、それぞれローカルメモリとして専用的に割り当てられている。

[0174] この場合、各CPUコアはローカルメモリにはダイレクトに(高速に)アクセスできるが、他のCPUパッケージに割り当てられたリモートメモリにアクセスするためには、当該CPUパッケージに属するCPUコアを経由する必要がある、その分、処理速度が低下する問題が生じる。

このため、このシステム310においては、各CPU コアがアクセスできるのはローカルメモリのみに限定され、他のCPUパッケージの配下にあるリモートメ

メモリへのアクセスは禁止されている。

[0175] このような割当て及び制約は、OS起動時に次のような「NUMAコントロール」のコマンドを入力することにより、実現される。

[0176] [1] APサーバ314に関する割当て

```
#numactl
    -physcpubind=1-15    ←コア(01)～(15)の割当て
    -membind=0          ←第1のメモリの割当て
    nice -n -5           ←プロセスの優先度の指定
    java -jar aloha.jar  ←APサーバプロセス（Java／登録商標）の
起動
```

[0177] [2] 第1のDBサーバ316に関する割当て

```
#numactl
    -physcpubind=17-23   ←コア(17)～(23)の割当て
    -membind=1           ←第2のメモリの割当て
    nice -n -5           ←プロセスの優先度の指定
    su -postgres         ←プロセスの実行ユーザをpostgresに切り
替え (su)
```

```
    -c "pg_ctl -D data1 start" ←第1のDBサーバプロセスの起動
(データ領域data1に対するデータベース) 」
```

[0178] [3] 第2のDBサーバ317に関する割当て

```
#numactl
    -physcpubind=24-30   ←コア(24)～(30)の割当て
    -membind=1           ←第2のメモリの割当て
    nice -n -5           ←プロセスの優先度の指定
    su -postgres         ←プロセスの実行ユーザをpostgresに切り
替え (su)
```

```
    -c "pg_ctl -D data2 start" ←第2のDBサーバプロセスの起動
(データ領域data2に対するデータベース) 」
```

[0179] [4] 第3のDBサーバ318に関する割当て

```
「#numactl
```

```
-physcpubind=31-38    ←コア(31)～(38)の割当て
```

```
-membind=2            ←第3のメモリの割当て
```

```
nice -n -5            ←プロセスの優先度の指定
```

```
su -postgres          ←プロセスの実行ユーザをpostgresに切り
```

替え (su)

```
-c "pg_ctl -D data3 start" ←第3のDBサーバプロセスの起動
```

(データ領域data3に対するデータベース) 」

[0180] [5] 第4のDBサーバ319に関する割当て

```
「#numactl
```

```
-physcpubind=39-45    ←コア(39)～(45)の割当て
```

```
-membind=2            ←第3のメモリの割当て
```

```
nice -n -5            ←プロセスの優先度の指定
```

```
su -postgres          ←プロセスの実行ユーザをpostgresに切り
```

替え (su)

```
-c "pg_ctl -D data4 start" ←第4DBサーバプロセスの起動
```

(データ領域data4に対するデータベース) 」

[0181] [6] 第5のDBサーバ320に関する割当て

```
「#numactl
```

```
-physcpubind=46-53    ←コア(46)～(53)の割当て
```

```
-membind=3            ←第4のメモリの割当て
```

```
nice -n -5            ←プロセスの優先度の指定
```

```
su -postgres          ←プロセスの実行ユーザをpostgresに切り
```

替え (su)

```
-c "pg_ctl -D data5 start" ←第5のDBサーバプロセスの起動
```

(データ領域data5に対するデータベース) 」

[0182] [7] 第6のDBサーバ321に関する割当て

```

「#numactl
  -physcpubind=54-60      ←コア(54)～(60)の割当て
  -membind=3              ←第4のメモリの割当て
  nice -n -5              ←プロセスの優先度の指定
  su -postgres            ←プロセスの実行ユーザをpostgresに切り
                           替え (su)
  -c "pg_ctl -D data6 start" ←第6のDBサーバプロセスの起動
                           (データ領域data6に対するデータベース)」

```

[0183] APサーバ314の業務処理部340、DB制御部342、第1のDB連絡部343～第10のDB連絡部352は、上記のように専用のプログラムに従い、CPUコア(01)～コア(15)が所定の処理を実行することにより実現されるのであるが、この際、OSやミドルウェアによって複数のスレッドが起動されて各CPUコアに割り当てられることにより、マルチタスク処理が実現される。

[0184] 図20は、各CPUコアと複数のスレッド378との対応関係を表しており、各スレッド378にはスレッドプール380が設けられ、そこに配置された複数のタスク382をスレッド378が順次処理していくイメージが描かれている。

この図におけるスレッド378が、APサーバ314の業務処理部340、DB制御部342、第1のDB連絡部343～第10のDB連絡部352として機能し、これらの各機能構成部が実行する具体的な処理が、タスク382に相当する。

[0185] 各スレッド378は、スレッドプール380に蓄積されたタスク382を古い順に次々と実行していき、自己のスレッドプール380が空になった場合には、他のスレッド378のスレッドプール380に蓄積されたタスク382を、新しい順に実行していく。

[0186] 図21に示すように、第1のDB連絡部343～第6のDB連絡部348と、第1のDBサーバ316～第6のDBサーバ321間は、各DBサーバに割り当てられたCPUコア数に対応した本数のコネクションプール384を介して接続されている。

例えば、第1のDBサーバ316には7個のCPUコアが割り当てられているため7本のコネクションプール384が設けられ、各コネクションプール384は予め

第 1 のDBサーバ316の特定のCPUコアと関連付けられている。

また、第 5 のDBサーバ320には 8 個のCPUコアが割り当てられているため 8 本のコネクションプール384が設けられ、各コネクションプール384は予め第 5 のDBサーバ320の特定のCPUコアと関連付けられている。

[0187] 補助記憶装置374内には、例えば、仕入テーブル386、注文テーブル387、注文明細テーブル388、注文取消テーブル389、品目テーブル390、完成品テーブル391、原材料テーブル392、原材料目録テーブル393、原価テーブル394、供給者テーブル395、顧客テーブル396等を含むテーブル群が、各DBサーバ単位で複数セット格納されている。

[0188] ここでは、第 1 のDBサーバ316～第 6 のDBサーバ321が設けられているため、A～Fの 6 セットのテーブル群が、それぞれ各DBサーバ専用として補助記憶装置374内の別個の領域に格納されている。

また、後述のように、各テーブル群に属する同一名称のテーブルには、それぞれ同一のレコードが同時に重複して格納される。

[0189] 図示は省略したが、第 7 のDBサーバ322～第 10 のDBサーバ330の補助記憶装置内にも、それぞれ仕入テーブル386、注文テーブル387、注文明細テーブル388、注文取消テーブル389、品目テーブル390、完成品テーブル391、原材料テーブル392、原材料目録テーブル393、原価テーブル394、供給者テーブル395、顧客テーブル396等を含むテーブル群が設けられており、第 1 のDBサーバ316～第 6 のDBサーバ321と同一内容のデータが格納される。

[0190] 図 22 は、各テーブルに格納されているレコードの構成を例示するものである。

すなわち、仕入テーブル386には、「仕入ID」、「原価ID」、「数量」のデータ項目を備えたレコードが格納されている。

また、注文テーブル387には、「注文ID」、「顧客ID」、「注文日時」のデータ項目を備えたレコードが格納されている。

注文明細テーブル388には、「注文明細ID」、「注文ID」、「品目ID」、「数量」のデータ項目を備えたレコードが格納されている。

注文取消テーブル389には、「注文明細ID」、「取消日時」のデータ項目を備えたレコードが格納されている。

[0191] 品目テーブル390には、「品目ID」のデータ項目を備えたレコードが格納されている。

完成品テーブル391には、「品目ID」のデータ項目を備えたレコードが格納されている。

原材料テーブル392には、「品目ID」のデータ項目を備えたレコードが格納されている。

原材料目録テーブル393には、「原材料目録ID」、「親品目ID」、「子品目ID」、「数量」のデータ項目を備えたレコードが格納されている。

原価テーブル394には、「原価ID」、「供給者品目ID」、「原価」のデータ項目を備えたレコードが格納されている。

供給者テーブル395には、「供給者品目ID」、「供給者ID」、「品目ID」のデータ項目を備えたレコードが格納されている。

顧客テーブル396には、「顧客ID」のデータ項目を備えたレコードが格納されている。

[0192] 上記の各テーブル386～396に関しては、以下の(1)～(4)の制約が予め課せられている。

(1)キー項目は一つに限定される。

したがって、複数の項目の組合せによってキー項目を構成すること（複合キー）は、禁止される。

[0193] (2)APサーバ314が各テーブルにアクセスするに際しては、レコードの参照と追加のみが許容され、既存レコードの削除と更新は禁止される。

[0194] このため、既存データを無効にする必要が生じた場合には、レコードを削除する代わりに、他のテーブルに新たなレコードを追加することによって削除と同等の状態が実現される。例えば、注文の取り消しがなされた場合、一般的には取消の対象となる注文データに「取消」を示す値を記述することで状態が管理されることになるが、このシステム310においては、注文取消テ

ブル389に対象となる「注文明細ID」のレコードを追加することで「取消」の事実が表現される。

[0195] また、既存レコードの値に変更が生じた場合には、修正済みの値を備えた異なるプライマリーキーのレコードを同一テーブルに追加することで、実質的に値の更新が実現される。例えば、原価テーブル394に格納されたあるレコードの原価を「300円」から「500円」に変更する場合、原価を「500円」とした新たなレコードが原価テーブル394に追加される。各レコードにはミリ秒精度のタイムスタンプが刻印されているため、これを比較することでレコード間の新旧（更新の事実）が明確化される。

[0196] このようにAPサーバ314によるレコードの削除及び更新が禁止されているため、各テーブルには過去の如何なる時点のデータも温存されることとなり、将来において変更履歴の再現が必要となった場合にも対応できる利点がある。

ただし、十分に時間が経過し、データ保持の必要性が完全になくなった場合には、各DBサーバに直接ログインして削除のSQLを発行することにより、レコードを削除することはできる。

[0197] (3)各テーブルのカラムには、NULL値禁止制約、フラグ値禁止制約、区分値禁止制約が課せられる。

まず「NULL値禁止制約」とは、データ項目の値としてNULL（値なし）を充填することが禁止されることを意味しており、このようなNULL値の充填を想定したデータ項目の設定自体が許容されないことになる。

つぎに「フラグ値禁止制約」とは、データ項目の値としてフラグ値（「1／0」、「ON／OFF」、「TRUE／FALSE」等の2値データ）を充填することが禁止されることを意味しており、このようなフラグ値の充填を想定したデータ項目の設定自体が許容されないことになる。

「区分値禁止制約」とは、データ項目の値として区分値（「1：正社員」、「2：パート」、「3：アルバイト」等）を充填することが禁止されることを意味しており、このような区分値の充填を想定したデータ項目の設定自

体が許容されないことになる。

[0198] (4)機能の拡張に際し、既存テーブルにカラムを追加する代わりに、テーブルの新設で対応する。

すなわち、上記のように「注文明細テーブル388」に格納された注文を取り消すに際し、現状では「注文取消テーブル389」に取り消したい注文の注文明細IDを登録することによって対応しているため、図23(a)に示すように、注文自体は維持したまま数量のみを減じたい場合であっても、一旦、全数量について取り消されてしまう。このためユーザは、減らした数量で再度発注する必要がある。

[0199] ここで、注文を維持したまま数量のみを減じることができるよう機能を拡張するとなると、図23(b)に示すように、注文取消テーブル389に「取消数量」のカラムを追加することが考えられる。この結果、確かに注文を維持したまま数量のみを減じるという目的は達成できるが、多数のDBサーバにおいて現に使用中のテーブルに対して一斉に改変を加えるのは容易ではなく、また新機能に対応させたアプリケーションプログラムに不具合が生じた等の理由で元の機能に戻す必要が生じた場合でも、迅速に対応できないという問題が生じる。

[0200] このため、このシステム310では図24に示すように、従来の注文取消テーブル389は残したまま、「注文明細ID」、「取消数量」、「取消日時」のデータ項目を備えた部分取消テーブル389'を各DBサーバ内に新設する方式を採用している。

すなわち、数量単位での部分取消を認める新サービスに切り替わった時点で、注文取消テーブル389はシステム310から切り離される一方、注文の取消は部分取消テーブル389'に対象となる注文明細ID及び取消数量を備えたレコードを追加することによって記録される（注文自体を取り消す場合には、注文の全数量が「取消数量」として設定される。）

[0201] このように、多数のDBサーバにおいて現在稼働中のテーブルに対して一斉に修正を加えることに比べ、各DBサーバに新規のテーブルを追加の方が容

易に実現できる。また、機能拡張前のテーブル（注文取消テーブル389）は改変されることなく温存されるため、万一、サービスを元に戻す必要が生じた場合にも迅速に対応できる利点がある。

[0202] 上記(1)～(4)の制約ルールは、新規テーブルの設計時やSQL文発行時に、各DBサーバのデータベース管理システムによって適否がチェックされ、制約ルールに違反する処理の実行が拒絶されることにより、その実効性が担保される。

[0203] このシステム310の場合、各テーブルに格納される個々のレコードが上記の通り極限まで正規化されており、またデータ間の演算処理は、後述のようにAPサーバ314のDB制御部342によって行われ、DBサーバ側の演算処理によって冗長な中間データが生成されることもないため、レコードの削除や更新の代わりに新規レコードの挿入によって対応するとしても、ディスク容量が従来に比べて徒に圧迫されることはない。また、キャッシュメモリのヒット率が向上するため、DBサーバの処理の高速化にも資する。

[0204] このようなテーブル群によって業務データが管理されている状況下において、新規データの登録を希望するユーザは、業務処理部340からクライアント端末336に送信されたデータ入力画面（図示省略）を通じて、データを入力する。

[0205] この入力データを業務処理部340から受け取ったDB制御部342は、第1のDB連絡部343～第10のDB連絡部352に対してデータの登録を依頼する。

各DB連絡部は、この入力データの登録を依頼するSQL文を、それぞれ自己の担当するDBサーバに発行する。

[0206] これを受けた各DBサーバでは、対応のテーブルに対するレコードの挿入が一斉に実行される。

この結果、各DBサーバが独立して管理している同名テーブルに、同一レコードが重複して格納されることとなる。

[0207] なお、第1のDB連絡部343～第6のDB連絡部348は、それぞれが担当する各DBサーバ（第1のDBサーバ316～第6のDBサーバ321）に対し、特定のCPUコア

に関連付けられたコネクションプール384を通じてSQLを発行することにより、各DBサーバにおける担当CPUコアが指定される。

これを受けた各DBサーバでは、指定されたCPUコアにより、対応のテーブルに対するレコードの挿入が一斉に実行される。

[0208] 各DBサーバは、DB連絡部からデータを受け取って自身のメモリに格納した時点で、DB連絡部に対して受取完了通知を送信し、その後、ハードディスクやSSD等の補助記憶装置内に設けられたテーブルに対応のレコードを格納する。

[0209] このDBサーバから送信された受取完了通知は、各DB連絡部を経由してDB制御部342に集められる。

DB制御部342は、全てのDB連絡部から受取完了通知が返ってきた時点で、業務処理部340に対して登録完了通知を出力する。

[0210] 業務処理部340は、この登録完了通知をDB制御部342から受け取った時点で、上記データを読み込みの対象と認識する。逆に言えば、登録完了通知が返ってくるまでの間、業務処理部340は当該データを読み込みの対象外とすることで、各DBサーバにおける登録のタイミングがずれて不正なデータが読み込まれてしまう危険性を回避している。

[0211] このシステム310の場合、上記のように、各業務データは挿入のみが許容され、削除や更新が認められないルールが適用されているため、データの復旧が極めて容易となる。

すなわち、データの削除や更新が許容されることを前提とした場合、データを復旧させるためにはDBサーバが保持している更新履歴情報に基づき、一定の時点からデータの追加や削除、更新を順番に再現する必要がある、これに長い時間と大きな負荷を要することになる。また、更新ログを確実に保存する仕組みをDBサーバ側に設ける必要がある。

[0212] これに対し、このシステム310のようにデータの挿入のみが許容され、削除と更新が禁止される前提の場合、他の正常なDBサーバに登録されたデータと障害が発生したDBサーバに登録されたデータを比較し、単純に足りないデー

タをコピーするだけで済み、データの順序（先後関係）を考慮する必要がなくなる。

[0213] 例えば、サーバコンピュータ324自体が故障した場合には、同一構内に設置された第7のDBサーバ322内のデータを新たなサーバコンピュータ内に取り込むことで、極めて容易にデータの復旧が可能となる。

あるいは、地震等の大規模災害が発生し、データセンター312が壊滅的な打撃を受けた場合には、遠隔地に設置された第8のDBサーバ326～第10のDBサーバ330内のデータに基づいて、比較的早期にデータの復旧が可能となる。

[0214] このシステム310にあっては、上記のようにDB連絡部から送信されたデータがDBサーバのメモリに格納された時点で、DBサーバから受取完了通知がAPサーバ314に発行されるため、極めて短時間の中に業務処理部340は当該データを読み込みの対象とすることができる。

[0215] これまでの常識では、メモリは揮発性の記憶手段であり、電源供給が停止された時点でデータが喪失してしまうため、不揮発性の記憶手段（ハードディスク等）にデータが格納された時点で完了通知が返されるべきとなる。

これに対し、このシステム310の場合には、あるDBサーバでデータの欠損が発生しても、他のDBサーバから不足データをコピーするだけで追いつくことができるため、データがハードディスク等に格納されるまで待機する必要がない。

[0216] つぎに、ある完成品（例えば品目ID「NRI-0001」）の在庫数についてクライアント端末から参照リクエストが送信された場合の処理手順について、図25及び図26のフローチャートに従い説明する。

[0217] まず、業務処理部340から送信された検索条件指定画面（図示省略）上でユーザが完成品である品目ID「NRI-0001」の在庫検索を指定すると、この検索リクエストの実現に必要な処理が、業務処理部340によって特定される（図25のS310）。

[0218] 具体的には、以下の処理内容が特定された後、順次実行される。

(01)注文明細テーブル388から、品目ID「NRI-0001」を備えた全レコードを抽

出する（S312）。

(02)品目ID「NRI-0001」の累積注文数量を算出する（S314）。

(03)注文取消テーブル389から、品目ID「NRI-0001」を備えた全レコードを抽出する（S316）。

(04)品目ID「NRI-0001」の累積注文取消数量を算出する（S318）。

(05)品目ID「NRI-0001」の累積注文数量から累積注文取消数量を減算して、現時点における累積実注文数量を算出する（S320）。

(06)原材料目録テーブル393を参照し、品目ID「NRI-0001」を親品目IDとして、いる子品目IDと、その数量を取得する（S322）。この子品目IDが、完成品である品目ID「NRI-0001」の原材料を示している。

(07)各子品目IDをキーにして、供給者テーブル395から供給者品目IDを取得する（図26のS324）。

(08)供給者品目IDをキーにして、原価テーブル394から原価IDを取得する（S326）。

(09)原価IDをキーにして、仕入テーブル386から当該原材料に係る全レコードを抽出する（S328）。

(10)原材料毎に、累積仕入数量を算出する（S330）。

(11)各原材料の累積仕入数量に基づいて、品目ID「NRI-0001」の生産可能数量を算出する（S332）。

(12)この生産可能数量から累積実注文数量を減算することにより、当該完成品の現在における在庫数量を算出する（図25のS334）。

(13)最後に、「NRI-0001」の在庫数量が記載された処理結果画面（図示省略）をクライアント端末336に送信する（S336）。

[0219] これらの処理の中、データの抽出処理は、第1のDB連絡部343～第6のDB連絡部348の何れかが、第1のDBサーバ316～第6のDBサーバ321の何れかに対してSQL文を発行することによって実行される。

この際、各DB連絡部は各DBサーバに対し、特定のCPUコアに関連付けられたコネクションプール384を通じてSQL文を送信する。この結果、DBサーバにお

けるレコードの抽出処理は、指定されたCPUコアによって実行される。

また、各DBサーバから送信されたデータに対しては、DB制御部342によってソートやマージ、マッチング、コントロールブレイク等の必要な演算処理が施される。

[0220] ところで、上記の各処理の中には、先行する処理が完了しないと後続の処理を実行できないものと、同時並行して処理が可能なものがある。

例えば、図25に示したように、注文明細テーブル388から該当品目IDのレコードを抽出するS312の処理と、注文取消テーブル389から該当品目IDのレコードを抽出するS316の処理と、原材料目録テーブル393を参照し、該当品目IDを親品目IDとしている子品目ID及び数量を取得するS322の処理は、品目IDが特定された時点で同時並行することができる。

このため、DB制御部342は、上記の処理をそれぞれ別個のCPUコアを指定し、対応のDB連絡部を通じて各DBサーバに抽出処理を依頼する。

[0221] また、S322の処理によって完成品を生産するのに必要な原材料の種類が特定された時点で、図26に示したように、各原材料の仕入数量を算出するのに必要なデータの抽出処理が開始されることになるが、このS324～S328も原材料毎に同時並行で処理される。

例えば、原材料の数が50種類であった場合、DB制御部342は、それぞれ異なる50個のCPUコアを指定し、対応のDB連絡部を通じて各DBサーバにレコードの抽出を依頼する。

このように、特定のCPUコアに具体的なレコードの抽出を直接的に割り振ることにより、効率的にデータの参照が可能となる。

[0222] なお、原材料の種類が数百種以上に及ぶ場合、原材料単位の処理を一度に異なるCPUコアに割り当てることができなくなる。

このような場合、DB制御部342は各CPUコアに順番に原材料毎の処理を依頼し、処理が完了した時点で当該CPUコアに残りの原材料単位の処理を順番に割り当ててを繰り返す。

[0223] 特定の業務処理を実現するために必要なデータの種類や抽出手順について

はAPサーバ314のアプリケーションプログラムが認識しているため、具体的なレコードの抽出を各DBサーバに対しCPUコアを指定した形で依頼することにより、マルチコアCPUの潜在能力を最も効果的に引き出すことが可能となる。

[0224] なお、OSには統計情報に基づいてデータ処理の効率化や省電力化を実現するため、処理の途中で取えてCPUのクロック数をダウンさせる機能が設けられているが、本システム310のようにCPUコア単位で処理内容をきめ細かく制御する際には邪魔になるため、この種の余計なサービスは事前に無効化しておくことが望ましい。

[0225] 上記のように、このシステム310の場合、各DBサーバには同一のデータが重複して蓄積されているため、DB制御部342は必要なデータを最も高速で処理可能なDBサーバから自由を取得することができるし、複数のDBサーバから分散的にデータを取得して結合させることもできる。

[0226] また、データの参照や登録に必要なSQL文は各DB連絡部によって発行されるため、業務処理部340やDB制御部342を実現するためのアプリケーションプログラム中にSQL文を記述する必要がなくなる。

しかも、各DBサーバに格納されたテーブルは極度に正規化されており、各DB連絡部はDB制御部342に指定されたデータの登録と抽出のみを担当するものであるため、DB連絡部用のプログラム中に記述されるSQL文も極めて簡素なもので間に合い、サービスの機能拡張に際して一々改変する必要がない。

もちろん、サービスの機能拡張時には業務処理部340及びDB制御部342を実現するためのアプリケーションプログラムについては修正を施す必要があるが、SQL文の記述を含まない分、見通しが良好となり、バグの発見が容易となる。

図面の簡単な説明

[0227] [図1]この発明に係るデータ管理システムの全体構成図である。

[図2]第1のAPサーバ及び第2のAPサーバの内部構成を示すブロック図である。

[図3]予約データの構造を示す図である。

[図4]人工キーの発行手順を示すフローチャートである。

[図5]人工キー管理テーブル等の構造を示す図である。

[図6]予約取消データの構造を示す図である。

[図7]レコードを実質的に更新する様子を示す図である。

[図8]一連の関連データを各DBサーバに送信する様子を示す図である。

[図9]一連の関連データを複数のコネクションを介してAPサーバからDBサーバへ並列送信する様子を示す図である。

[図10]CPUコアとスレッドとの対応関係を示す図である。

[図11]この発明に係る他のデータ管理システムを実現するためのハードウェア構成を示す模式図である。

[図12]この発明に係るデータ管理システムの機能構成を示すブロック図である。

[図13]APサーバの各CPUコアと複数のスレッドとの対応関係を示す模式図である。

[図14]各テーブルに格納されているレコードの構成例を示す図である。

[図15]このシステムにおける処理手順を示すフローチャートである。

[図16]このシステムにおける処理手順を示すフローチャートである。

[図17]この発明に係るサービス提供システムの全体構成図である。

[図18]APサーバの機能構成を示すブロック図である。

[図19]APサーバのハードウェア構成を示す模式図である。

[図20]APサーバの各CPUコアと複数のスレッドとの対応関係を示す模式図である。

[図21]APサーバの一部のDB連絡部と、APサーバと同一の筐体内に設けられた各DBサーバとの間の対応関係を示す模式図である。

[図22]各テーブルに格納されているレコードの構成例を示す図である。

[図23]サービスの機能拡張時におけるテーブル構成の変更例を示す図である。

[図24]サービスの機能拡張時におけるテーブル構成の変更例を示す図である。

。

[図25]このシステムにおけるデータ抽出時の処理手順を例示するフローチャートである。

[図26]このシステムにおけるデータ抽出時の処理手順を例示するフローチャートである。

符号の説明

- [0228] 110 データ管理システム
- 112 第1のデータセンター
- 114 第1のAPサーバ
- 116 第1のDBサーバ
- 118 第2のDBサーバ
- 120 第2のデータセンター
- 122 第2のAPサーバ
- 124 第3のDBサーバ
- 126 第4のDBサーバ
- 127 通信ネットワーク
- 128 クライアント端末
- 129 通信回線
- 130 業務処理部
- 132 人工キー管理部
- 134 データ制御部
- 138 DB連絡部
- 140 予約データ
- 141 予約統合データ
- 142 予約店データ
- 143 予約顧客データ
- 144 予約金額データ
- 145 予約年月日データ

- 150 予約統合テーブル
- 151 予約店テーブル
- 152 予約顧客テーブル
- 153 予約金額テーブル
- 154 予約年月日テーブル
- 156 人工キー管理テーブル
- 157 下一桁管理テーブル
- 158 サーバ管理テーブル
- 159 データセンター管理テーブル
- 160 取消統合データ
- 161 取消対象データ
- 162 取消年月日データ
- 163 取消統合テーブル
- 164 取消対象テーブル
- 165 取消年月日テーブル
- 166 予約取消データ
- 170～174 コネクション
- 180 CPUコア
- 181 スレッド
- 182 スレッドプール
- 183 タスク
- 210 データ管理システム
- 211 サーバコンピュータ
- 212 第1のパッケージ
- 214 第2のパッケージ
- 216 第3のパッケージ
- 218 第4のパッケージ
- 220 マザーボード

222	第 1 のメモリ
224	第 2 のメモリ
226	第 3 のメモリ
228	第 4 のメモリ
230	外部記憶装置
232	APサーバ
236	第 1 のDBサーバ
238	第 2 のDBサーバ
240	第 3 のDBサーバ
242	第 4 のDBサーバ
244	第 5 のDBサーバ
246	第 6 のDBサーバ
252	画面処理部
254	処理内容特定部
256	DB処理割当部
258	演算処理部
260	通信ネットワーク
262	クライアント端末
264	スレッド
266	スレッドプール
268	タスク
270	コネクションプール
272	仕入テーブル
273	注文テーブル
274	注文明細テーブル
275	注文取消テーブル
276	品目テーブル
277	完成品テーブル

278	原材料テーブル
279	原材料目録テーブル
280	原価テーブル
281	供給者テーブル
310	サービス提供システム
312	データセンター
314	APサーバ
316	第 1 のDBサーバ
317	第 2 のDBサーバ
318	第 3 のDBサーバ
319	第 4 のDBサーバ
320	第 5 のDBサーバ
321	第 6 のDBサーバ
322	第 7 のDBサーバ
324	サーバコンピュータ
326	第 8 のDBサーバ
328	第 9 のDBサーバ
330	第 10 のDBサーバ
332	通信回線
334	インターネット
336	クライアント端末
340	業務処理部
342	DB制御部
343	第 1 のDB連絡部
344	第 2 のDB連絡部
345	第 3 のDB連絡部
346	第 4 のDB連絡部
347	第 5 のDB連絡部

348	第 6 のDB連絡部
349	第 7 のDB連絡部
350	第 8 のDB連絡部
351	第 9 のDB連絡部
352	第 1 0 のDB連絡部
360	第 1 のCPUパッケージ
362	第 2 のCPUパッケージ
364	第 3 のCPUパッケージ
366	第 4 のCPUパッケージ
368	マザーボード
370	第 1 のメモリ
371	第 2 のメモリ
372	第 3 のメモリ
373	第 4 のメモリ
374	補助記憶装置
378	スレッド
380	スレッドプール
382	タスク
384	コネクションプール
386	仕入テーブル
387	注文テーブル
388	注文明細テーブル
389	注文取消テーブル
390	品目テーブル
391	完成品テーブル
392	原材料テーブル
393	原材料目録テーブル
394	原価テーブル

395 供給者テーブル

396 顧客テーブル

請求の範囲

[請求項1]

相互に所定の距離を隔てて設けられた複数のデータセンター内に、それぞれAPサーバと複数のDBサーバを設置したデータ管理システムであって、

上記の全DBサーバは、相互に共通するデータを格納するテーブルをそれぞれ備える共に、各テーブルにはレコードの削除及び更新が禁止される制約が設けられており、

上記の各APサーバは、同一データセンター内に設置された各DBサーバと接続されると共に、他のデータセンター内に設置された各DBサーバとも接続されており、

また各APサーバは、少なくとも一つの業務処理部と、人工キー管理部と、データ制御部と、複数のDB連絡部を備えており、

上記業務処理部は、業務処理の結果である業務データを生成するに際し、上記人工キー管理部に対して人工キーの発行を要求すると共に、この人工キー管理部によって発行されたユニークな人工キーよりなるIDを主キーとして備えた業務データを生成し、

上記人工キー管理部は、同一データセンター内に設置された人工キー管理テーブルを参照し、現時点で最新の人工キーの値を上記業務処理部に渡すと同時に、当該テーブルの値を更新させる処理を実行し、

上記データ制御部は、上記業務処理部から出力された業務データを複製して各DB連絡部に出力すると共に、各DB連絡部を介して各DBサーバからの受取完了通知が送信された際に、全DBサーバからの受取完了通知が揃っていることを確認した上で、上記業務処理部に登録完了通知を出力し、

上記の各DB連絡部には、それぞれ担当するDBサーバの特定情報が設定されており、

各DB連絡部は、上記データ制御部から業務データが出力された際に、自己の担当するDBサーバにSQLを発行してレコードの追加を求める

と共に、当該DBサーバから受取完了通知が送信された場合には、これを上記データ制御部に出力する処理を実行し、

さらに、上記の各人工キーの所定の位置には、APサーバの識別符号が設定されており、

上記業務処理部は、上記データ制御部から登録完了通知が渡されるまでは当該業務データの参照を控えることを特徴とするデータ管理システム。

[請求項2] 上記業務データは、上記業務処理部による業務処理の結果として生成された、相互に関連性を有する複数の実体データと、各データを束ねるために生成された統合データの組合せよりなり、

この統合データは、人工キーよりなる統合IDを主キーとして格納するデータ項目のみを備え、

上記の各実体データは、当該実体データを格納するデータ項目と、人工キーよりなる独自のIDを主キーとして格納するデータ項目と、上記統合データのIDを外部キーとして格納するデータ項目を備えていることを特徴とする請求項1に記載のデータ管理システム。

[請求項3] 上記の各APサーバと各DBサーバとの間が、それぞれ複数のコネクションを介して接続されると共に、

各APサーバにはコネクション単位で上記のDB連絡部が設けられており、

上記業務データは、各DB連絡部及びコネクションを介して並列的に各DBサーバに送信されることを特徴とする請求項1または2に記載のデータ管理システム。

[請求項4] 上記業務処理部は、既存の業務データを更新する必要がある場合に、当該業務データのIDを格納するデータ項目と、人工キーからなる独自のIDを主キーとして格納するデータ項目と、修正後の実体データを格納するデータ項目を備えた業務データを新たに生成し、

上記データ制御部及び各DB連絡部を介してこの業務データを各DBサ

サーバに設けられた既存のテーブルに追加登録させることにより、既存の業務データを実質的に更新することを特徴とする請求項1～3の何れかに記載のデータ管理システム。

[請求項5] 上記業務処理部は、既存の業務データを削除する必要がある場合に、当該業務データのIDを格納するデータ項目と、人工キーからなる独自のIDを主キーとして格納するデータ項目を備えたデータ取消用の業務データを生成し、

 上記データ制御部及び各DB連絡部を介してこのデータ取消用の業務データを各DBサーバに設けられた取消専用のテーブルに追加登録させることにより、既存の業務データを実質的に削除することを特徴とする請求項1～4の何れかに記載のデータ管理システム。

[請求項6] 複数のCPUコアを内蔵したCPUパッケージ、及び複数のテーブルを格納しておく外部記憶装置を備えたコンピュータと、

 このコンピュータ上でAPサーバ用プログラムを起動させることによって構成されたAPサーバと、

 上記コンピュータ上でDBサーバ用プログラムを複数起動させることによって構成された複数のDBサーバを備え、

 上記の外部記憶装置には、相互に共通するデータを格納する複数のテーブルが、DBサーバ単位で別個に格納されており、

 上記のAPサーバ及び各DBサーバには、それぞれ複数の異なるCPUコアが専用的に割り当てられており、

 上記APサーバは、業務処理の実行に必要なデータの種別を特定する機能と、

 上記DBサーバの何れかに対して必要なデータの抽出を依頼する機能と、

 当該DBサーバから取得したデータに対して必要な演算処理を施す機能と、

 業務処理によって生じたデータの登録を、上記の全DBサーバに対し

て同時に依頼する機能を少なくとも備えており、

上記DBサーバにデータの抽出または登録を依頼するに際しては、当該処理を担当するCPUコアを具体的に指定した状態でSQLを発行し、

これを受けたDBサーバでは、指定されたCPUコアが対応の処理を実行することを特徴とするデータ管理システム。

[請求項7]

上記コンピュータは、上記CPUパッケージを複数備えており、

各CPUパッケージには専用のローカルメモリが割り当てられており、

上記APサーバ及び上記DBサーバの少なくとも一部には、それぞれ別個のCPUパッケージに属するCPUコアが割り当てられており、

上記APサーバ及び各DBサーバには、ローカルメモリにのみアクセス可能で、他のCPUパッケージに専用的に割り当てられたローカルメモリにはアクセスできない制約が課せられていることを特徴とする請求項6に記載のデータ管理システム。

[請求項8]

上記APサーバには、上記の各DBサーバに対してデータの挿入及び抽出を依頼することのみが許容され、データの更新及び削除を依頼することが禁止される制約が課せられていることを特徴とする請求項6または7に記載のデータベース管理システム。

[請求項9]

APサーバと、このAPサーバと接続された複数のDBサーバとを備え、クライアント端末に対して所定の情報処理サービスを提供するサービス提供システムであって、

上記APサーバは、上記の各DBサーバ単位で設けられた複数のDB連絡部と、業務処理部と、DB制御部とを備え、

上記の各DB連絡部には、それぞれ担当するDBサーバの特定情報が設定されており、

上記の各DBサーバには、相互に共通するデータを格納するテーブルがそれぞれ設けられており、

各テーブルには、レコードの削除及び更新が禁止される制約が設け

られており、

上記業務処理部からデータの登録依頼を受けた際に、上記DB制御部は、当該データを複製して上記の全DB連絡部に渡し、

これを受けた各DB連絡部は、自己の担当するDBサーバにSQL文を発行してレコードの追加を求め、

上記業務処理部から特定データの参照を求められた場合に、上記DB制御部は、上記の各DBサーバの中から1または複数のDBサーバを特定し、当該DBサーバを担当しているDB連絡部に対してデータの抽出を依頼し、

これを受けたDB連絡部は、自己の担当しているDBサーバに対してSQL文を発行してレコードの抽出を依頼すると共に、DBサーバから受け取ったデータをDB制御部に渡し、

DB制御部は、受け取ったデータに必要な演算処理を施した上で、上記業務処理部に渡すことを特徴とするサービス提供システム。

[請求項10] 上記業務処理部は、既存データを更新する必要がある場合に、更新後の値を備えたデータを新たに生成し、

上記DB制御部及び各DB連絡部を介してこのデータを各DBサーバに設けられた既存のテーブルに追加登録させることにより、既存データの実質的な更新を実現することを特徴とする請求項9に記載のサービス提供システム。

[請求項11] 上記業務処理部は、既存の業務データを削除する必要がある場合に、当該業務データのIDを格納するデータ項目を備えたデータ取消用のデータを生成し、

上記DB制御部及び各DB連絡部を介してこのデータ取消用のデータを各DBサーバに設けられた取消専用のテーブルに追加登録させることにより、既存データの実質的な削除を実現することを特徴とする請求項9または10に記載のサービス提供システム。

[請求項12] 上記APサーバ及び上記DBサーバの中の少なくとも2以上のDBサーバ

が、同一のコンピュータ上に形成されており、

このコンピュータは、複数のCPUコアを内蔵したCPUパッケージと、複数のテーブルを格納しておく記憶装置を備え、

このコンピュータ上でAPサーバ用プログラムを起動させることによって上記APサーバが構成されると共に、このAPサーバ上で専用のアプリケーションプログラムを起動させることによって上記の業務処理部、DB制御部及び複数のDB連絡部が構成され、

同コンピュータ上でDBサーバ用プログラムを複数起動させることによって上記の各DBサーバが構成され、

上記APサーバ及び各DBサーバには、それぞれ複数の異なるCPUコアが専用的に割り当てられており、

上記DBサーバにデータの抽出及び登録を依頼するに際しては、上記DB連絡部から当該処理を担当するCPUコアを具体的に指定した形でSQL文が発行され、

これを受けたDBサーバでは、指定されたCPUコアによって対応の処理が実行されることを特徴とする請求項9～11の何れかに記載のサービス提供システム。

[請求項13]

請求項9に記載したサービス提供システムにおいて、クライアント端末に対して提供するサービスの機能を拡張するに際し、

拡張後の機能に対応したテーブルを上記の各DBサーバ内に追加すると共に、拡張前の機能に係るテーブルを各DBサーバ内に置いたまま処理の対象外に設定し、

上記APサーバの業務処理部、DB制御部及び各DB連絡部を、追加したテーブルに対応した拡張バージョンに置き換えることを特徴とするサービス提供システムの機能拡張方法。