

第Ⅲ章 オンライントランザクション処理 システム

第Ⅲ章 オンライントランザクション処理システム

学習目標

1. 集中システム環境、分散システム環境において、オンライントランザクション処理システムのソフトウェア構造について理解させる。
2. トランザクション処理モニタについて理解させる。
3. オンラインデータベース管理システムについて理解させる。
4. オンライントランザクション処理システムの応答時間を決める要因について理解させる。
5. オンライントランザクション処理システムで用いられている高性能化技術、高信頼化技術について理解させる。

全体概要

鉄道や航空券の座席予約、コンサートや演劇会のチケット予約、ホテル宿泊予約、銀行口座取引など、トランザクション処理は情報化社会に不可欠な存在である。地理的にも時間的にも空間的にも離れている複数の情報機器端末から刻々と入ってくる要求に対して、全体的にもまた個別にも矛盾のないデータ処理が行われる基本的なフレームワークを与えるのがトランザクションシステムである。

トランザクション処理の歴史は古く、汎用コンピュータにおいては相当な活躍をはたしてきたシステムである。これに要求される即時性や信頼性を永遠に向上させるためにオンライントランザクションシステムの改良が重ねられてきたし、ダウンサイジング時代に入っても専用のOLTPマシンが登場するなど低価格化と、タンデムコンピュータなどにみられるように高信頼化を勝ち得てきた。

最近顕著であることには、分散処理時代をくぐり抜け、クライアントサーバ時代に入りますます洗練されたオンライントランザクションシステムの構築に向けて努力が続けられていることである。

真のオープンシステム化時代を迎えつつある現在、メインフレームにおけるトランザクション処理とともに、ネットワークコンピューティングにおいてどのようにシステム構成が変わりつつあるかについても見通しておく必要がある。

内容のあらまし

節 項	内 容
<p>1. オンライントランザクション処理</p> <p>(1) オンライントランザクション処理</p> <p>(2) トランザクション</p> <p>(3) オンライントランザクション処理のハードウェア構成</p> <p>(4) オンライントランザクション処理のソフトウェア構成</p> <p>(5) 応答時間 (レスポンスタイム)</p> <p>(6) 高性能化と高信頼化</p> <p>(7) システム性能の評価</p>	<p>オンライントランザクションとは 原子性、一貫性、分離性、接続性 開始、作業、終了</p> <p>集中型トランザクション処理システム 分散型トランザクション処理システム</p> <p>画面制御、端末AP、通信処理、TPモニター 業務AP、システム管理、運転管理 障害管理</p> <p>通信時間、TP内待ち時間、業務処理時間 データベースアクセス時間</p> <p>ハードウェアによる高性能化と高信頼化 ソフトウェアによる高性能化と高信頼化</p> <p>TPC、TPC - A、TPC - C</p>
<p>2. オンライントランザクション処理のオペレーティングシステム</p> <p>(1) トランザクション処理モニター</p> <p>(2) 協調分散機能を持つトランザクション処理モニター</p>	<p>トランザクション処理システムのソフトウェア構成と処理の流れ</p> <p>負荷分散機能、トランザクションのルーティング機能 分散リソースのアクセス</p>
<p>3. オンラインデータベース</p> <p>(1) オンラインデータベース管理</p> <p>(2) マルチスレッド</p> <p>(3) 分散データベース</p> <p>(4) バックアップとリカバリ</p>	<p>データ定義機能、データ操作機能、データ制御機能</p>
<p>4. プログラムの構造</p>	<p>マルチタスク、プログラム域とデータ域の分離 メモリの節約</p>
<p>5. 分散型トランザクション処理の利点</p>	
<p>6. 主要用語</p>	

1. オンライントランザクション処理

(1) オンライントランザクション処理

日常生活において、鉄道の指定席券、航空券、演劇やコンサートのチケットの予約、銀行預金の引き出し、残高照会、銀行振込など、また、企業間での受発注においては、端末装置とホストコンピュータとが通信ネットワークを介して取引が行われる。このような取引は、情報処理の世界ではオンライントランザクション処理と呼ばれ、端末からの要求に対してホストコンピュータのプログラムは直ちに対応処理を行う。

例えば、鉄道の座席予約においては、顧客の希望する列車の予約状況が確認され空席であれば予約済みになり、発券が行われる。顧客が禁煙席を希望するか否か、あるいは隣合う席を希望するか等は予約端末から予約条件として付加される。予約条件としては、第1希望、第2希望なども添えることができる。例えば、コンサートなどのチケット予約では、S席が満席であれば、A席を、それもだめであればB席をとようになる。

このような処理を行うためにプログラム側では、要求内容に応じてデータベースをアクセスしてデータの参照、更新あるいは書き込みを行う。要求は、多数の端末から同時に発せられ、即座に処理されなければならない。データベースへの一連の処理で実行制御が行われ、この実行の単位をトランザクションと呼んでいる。

オンライントランザクション処理では、データベースは集中して管理される場合と、データの種類などに応じて分散して管理される場合が考えられる。どちらをとるかは、コスト、性能、信頼性などシステム要件によって決定される。

オンライントランザクション処理についてシステムの特徴をとらえると次のとおり。

- ・ 企業の基幹業務システムにおいてデータの更新を主体とした定型業務に適用される。
- ・ 受発注システム、在庫管理システム、製造管理システム、予約システム、バンキングシステムなどに代表される。
- ・ データベースに対する頻繁な参照、更新が行われる。
- ・ 各要求に対しては、数秒でレスポンスし、件数/秒というような単位時間当たりの一定の件数の処理が求められる。
- ・ 通信、データベースアクセス上のトラブルに対する確実なリカバリ処理が求められる。
- ・ システム構成として、大容量データベース、かなり多くの端末数、WAN、LAN、処理コンピュータ、システム管理の専門要員が必要である。

データベースのアクセスを必ずともなうことになるが、通常のデータベースアプリケーションが意思決定支援システムなどの非定型業務に適用され、データベースに対しては問い合わせが主体であり、レスポンスやリカバリがさほど重要でない点は対照的である。

なお、バッチ処理や第5章で説明されるリアルタイム処理との違いを簡単に整理しよう。バッチ処理は、複数のプログラムを一括して処理するが、大量のデータに対して大きな処理を行う点および即時処理を必要としない点が異なっている。また、リアルタイム処理は、原子力プラントのコントロール、ロケットの制御など実時間性が求められる点に違いがある。トランザクション処理は原子性（後述）、高速で予測可能な応答性、イベント起動型の処理という点で、バッチ、タイムシェアリングシステムと区別している。

オンライントランザクション処理システムに求められる要件として、ACID特性があげられる。四つの属性の頭文字をとって命名された用語である。

① 原子性 (Atomicity)

トランザクションはデータベースへの一連の入出力操作からなるが、データベースはトランザクション処理のすべての処理を完了した後の状態か、処理を受け付ける前の状態を保っているかどちらかでなければならない。

② 一貫性 (Consistency)

トランザクションがどのような順序で処理されようと、処理の結果が一致していなければならない。データベースが複数のテーブルからなっている場合でも各々のテーブル間で更新の同期が保たれる必要がある。

③ 分離性 (Isolation)

複数のトランザクションが同時に処理されている場合、単独で実行されたときと同一の処理結果を得なければならない。即ちお互いの処理に干渉し合わない。

④ 持続性 (Durability)

トランザクションとして完了してしまえば、その後システムに障害が起こっても更新されたデータベースの内容は保たれる。このACIDが保証されなければ、トランザクションがサポートされたとみなされないことになる。

(2) トランザクション

預金の引き出し、電子的な伝票の決済、在庫の問い合わせなどは、典型的なトランザクションである。これらをコンピュータシステムで見れば、

- ・ 端末からのホストへの要求の送信と受信した処理結果の表示に関連するユーザ・インタフェース処理
- ・ ホスト上のデータベースの参照、更新処理
- ・ 通信処理

の一連の処理ととらえられる。

トランザクション処理は、一連の要求をいくつかの作業単位に分割して実行するという考え方がベースとなっている。各作業単位は明示的に開始と終了を宣言する。トランザクションは、要求される作業が完全に終了する (コミット) か、その作業を中止する (ロールバック) かの決定がなされるまでの一次的な性格を持つ。

トランザクションを実現するプログラム手続きは以下のようなようになる。

- ・ 作業の開始 例え、指定席券の予約要求が発せられた時点
- ・ 作業 空き状況の確認と予約のためのデータベースアクセス
- ・ 端末側での 発券処理
- ・ 作業の終了 終了確認

(3) オンライントランザクション処理のハードウェア構成

従来は大型の汎用コンピュータを中核とした集中型のトランザクションシステムが作られてきたが、分散処理システムで構築される分散型トランザクションシステムも増えてきている。分散システムの場合、OLTP専用機によるものと、C/Sコンピューティングにより構成するものに分かれる。C/S型のシステムでは、取引の要求を発す端末側をクライアント、取引処理業務を行う側をサーバと呼ぶ。

集中型のトランザクション処理システムは、大規模で広域的にサービスを提供でき、そのためにデータベースの管理や業務処理は大型のコンピュータで行われる。全銀協システム、社会保険庁システムに代表される。

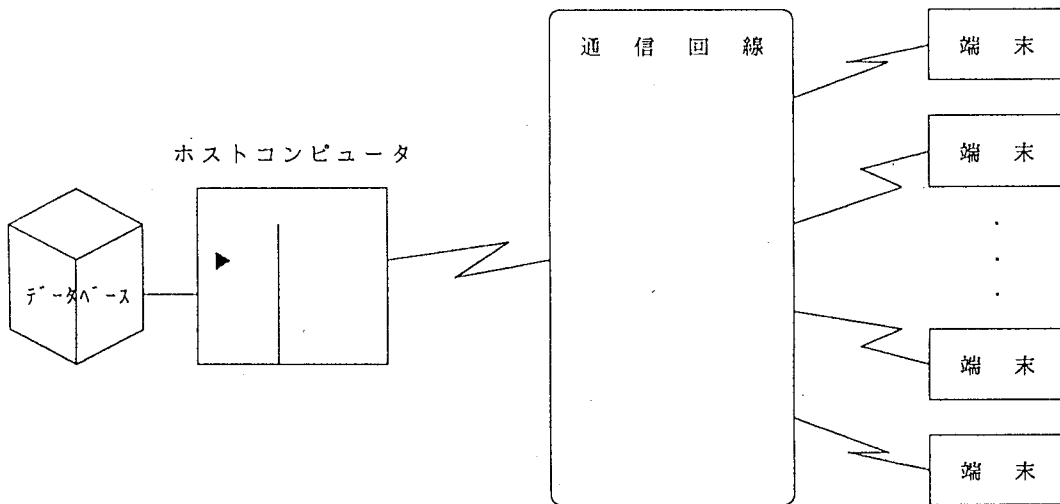
分散型のトランザクション処理システムは、企業や行政の基幹業務に適用され、クライアント/サーバ処理で実現される。データだけでなく、業務も複数のサーバに分散されることもありえる。地方行政情報システム、営業支援システムが代表的な例である。

- ① 集中型トランザクション処理システムでは、通常メインフレームコンピュータが中央にありすべてのデータベースはそこで集中的に管理され、各遠隔端末からのトランザクション要求が処理される。大規模な外部メモリと応答性を保証する主メモリ量を持ったシステム構成となっ

ている。ホスト側には、通信回線を介して端末とのデータ送受信および通信プロトコルを制御する通信制御装置が必要となる。

広域の通信回線網としては、公衆網、専用回線、DDX、ISDNが用いられる。また、端末は、従来、業務に専用特化したキャッシュディスペンサ、発券装置が主流であったが、最近では手元のインテリジェンシーを高めるためにパソコンが用いられる場合が多い。

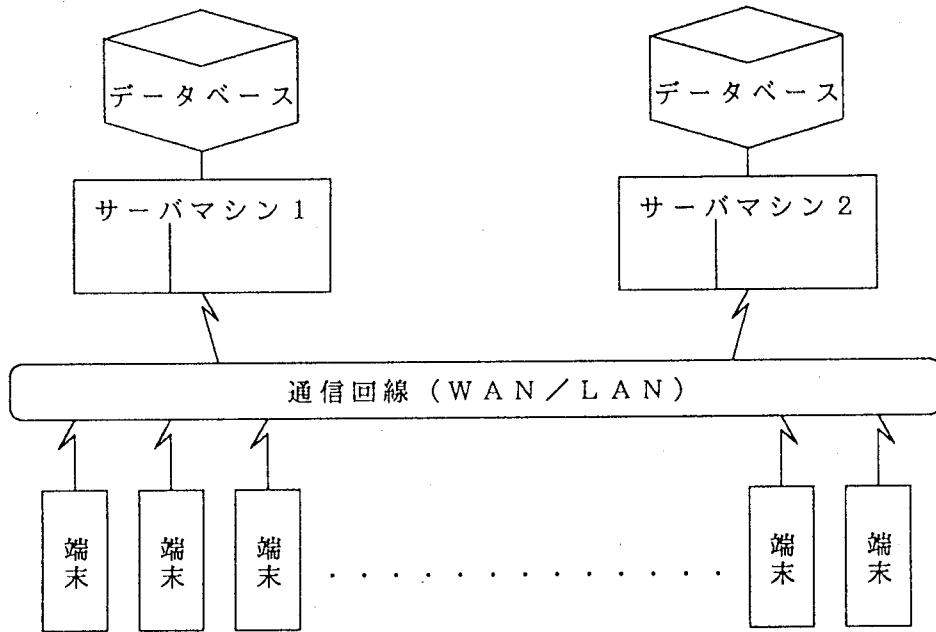
図表Ⅲ-1 集中型トランザクション処理システムのハードウェア構成



- ② 分散型トランザクション処理システムでは、取引処理マシンとして、オフィスコンピュータ、ワークステーション、トランザクション処理専用プロセッサが用いられる。この場合、処理の量やシステムの信頼性を考慮して複数台のマシンでサーバを構成することが多い。通信回線としては、分散マシン構内の処理という特徴もありLANが持ちられるが、遠隔とのデータ送受信もあり、その場合広域網が用いられる。

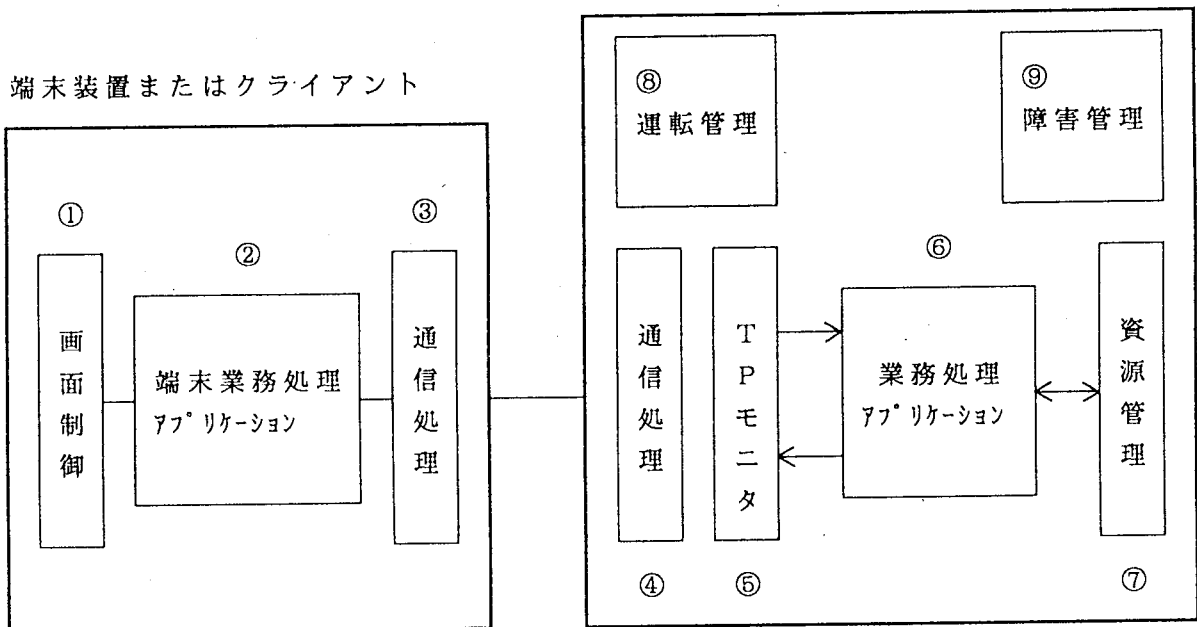
端末装置は、ワークステーションやパソコンによる場合が一般的で、集中型のシステムではホストコンピュータで行っていた業務の一部をそこで処理することができる。特に、GUI機能を生かす処理はほとんど端末側で行われる。

図表Ⅲ-2 分散型トランザクション処理システムのハードウェア構成



(4) オンライントランザクション処理のソフトウェア構成

オンライントランザクション処理は、元を正すと、遠隔地からの入出力による処理要求時点での即時処理（いわゆるオンラインリアルタイムシステム）と、種々の業務に応じて蓄積されるデータを一元的に管理し、全体の処理効率を高めようとするデータベースとを統合した形態である。当然データの共用や保全が図られる。



- ① 画面制御
端末の画面を介して処理要求を入力し、処理結果を表示するための制御である。
- ② 端末業務処理アプリケーション
端末業務処理においては、入力データの正確性が保証されなければならない。入力データのチェックは、当然センターマシンで行われるが、端末サイドでも負荷分散の意味でも相当行われなければならない。利用者自身の目視チェックや端末側のプログラムにおいて発生時点でのローカルなデータの矛盾が指摘され、訂正される。
- ③ 通信処理（端末側）
中央コンピュータとのコミュニケーションを司る。
- ④ 通信処理（中央側）
接続されている端末側とのコミュニケーションを司る。
- ⑤ TP モニタ：トランザクションを管理する。
- ⑥ 業務処理アプリケーション
集中型の場合ホストで、クライアントサーバ型の場合サーバで業務処理を行う。
- ⑦ 資源管理：データベースを中心とした資源へのアクセス
- ⑧ 運転管理：オンライントランザクションシステムの運用
- ⑨ 障害管理：システムダウン、データベース障害発生時、トランザクションがデータベースを更新途中であるとき、データベースが完全に復旧するまでの間当該データへのアクセスを抑制するような管理を行う。

(5) 応答時間（レスポンスタイム）

端末からトランザクション要求を送信してから、中央より処理結果を受信するまでの経過時間のこと。応答性に影響する要因としては以下が挙げられる。

- ① 通信時間
通信回線の速度、回線の混雑度
- ② TP モニタ内での待ち時間
トランザクションが輻輳した場合のスケジューリング待ち、資源使用における待ち、モニタのオーバヘッドなど。
- ③ トランザクション業務処理に要する計算処理を中心とした時間
- ④ データベースアクセス時間
データベース同時アクセスに関する待ちなど。

(6) 高性能化と高信頼化

- ① 高性能化
 - a. ハードウェアによる高性能化
 - ア. CPU 能力の向上
マルチプロセッサ化、並列処理プロセッサ化
 - イ. ディスク装置の能力向上
並列アクセス、ディスクキャッシュ
 - ウ. 通信能力向上
 - b. ソフトウェアによる高性能化
 - ア. 処理実行単位の優先度制御
マルチスレッド技術
 - イ. 多重並行処理化
 - ウ. データベースアクセスのチューニング

② 高信頼化

a. ハードウェアによる高信頼化

ア. RAID (Redundant Arrays of Inexpensive Disk)

イ. フォールトトレラントマシンの採用

汎用大型機のトランザクション処理システムでの故障対策としては、デュプレックスシステムやデュアルシステムが用いられてきたが、完璧な復旧は非常に困難であり、また、負荷と費用も膨大であることから、フォールトトレラントマシンの有効性がクローズアップされてきた。

このマシンではいかなる故障であっても、1箇所だけであればシステムの運転を継続できること、また、システムの他の部分に影響を与えずに故障を修復できることが特徴である。

ウ. プロセッサの多重化

エ. 無停電電源装置の利用 (UPS)

b. ソフトウェアによる高信頼化

ア. ログやジャーナルの機構とそれに基づくリカバリ処理

ロールバック、リカバリ、ロールフォワードを行うために、TPシステムにより処理される全てのトランザクションに関するアクションシーケンスをログ情報として専用の安定した記憶装置上に保存する。

障害の発生後には、素早くロールバック、リカバリ、ロールフォワードが行われなければならない。

イ. ディスク内容のバックアップ

リカバリ情報が格納されているディスクがクラッシュすると、障害からの十分なリカバリ手段はない。ロールバックができるようにするためには、データとログファイルを複製して、一度には障害が発生しない別々のディスクコントローラに接続されているディスク上に格納することが必要である。

ウ. ミラーディスク機能

データのコピーが存在するペアのディスクを二重化ディスクもしくはミラーディスクと言う。

エ. 障害検出と処理再開機能

オ. 現用/予備系の自動切り替え機能

(7) システム性能の評価

トランザクション処理システムに対する性能評価基準を決める標準化団体であるTPC (Transaction Processing Council: トランザクション処理性能評議会) の定めた性能値が参考にされる。

コンピュータ各社は、製品のトランザクション処理性能を公表している。クライアントサーバ型トランザクション処理では事例が増えている。

① TPC-A

TPCの最初の標準ベンチマークである。銀行におけるATMに見られるような入出金システムを想定した処理仕様となっている。業界では性能評価の手法として広く普及している。今ではこの仕様が製品の生き残り戦略になるほど競争の手段と化している。

② TPC-C

TPC-Aを補完するためのベンチマーク。TPC-Aが比較的単純な業務処理を想定して測定された。実際の業務処理にはずっと複雑なアプリケーションが増えてきており、TPC-Aだけでは処理能力の判定としては不十分な面が出てきた。

そのために新たに開発されたのがTPC-Cであり、これは受発注業務を例に採った実ジョブをモデルに負荷を設定した。

2. オンライントランザクション処理のオペレーティングシステム

通常のユーザアプリケーションは、オペレーティングシステムの制御下で管理され動作する。オンライントランザクション処理では、各トランザクションは直接オペレーティングシステムの制御下に入るのではなく、トランザクション処理モニタの管理下で動作が制御される。オペレーティングシステムでは、タスクという概念が管理の対象、トランザクション処理ではトランザクションという概念が管理を受けることになる。トランザクション処理モニタは、オペレーティングシステムの制御下では通常のタスク的な位置づけにある。

(1) トランザクション処理モニタ

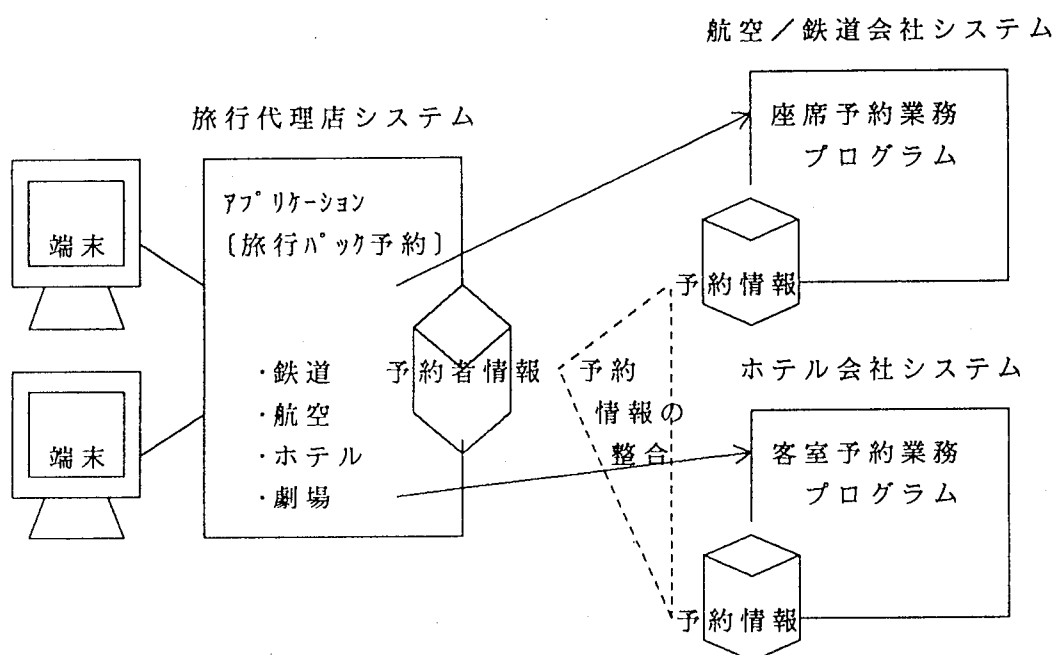
処理形態で分類すると、集中型のトランザクション処理システムモニタと分散型のトランザクション処理システムモニタとがある。

① トランザクション処理システムのイメージ

まず、トランザクション端末利用者から見た場合、ソフトウェアとしては共通データベースへのアクセスとそれらのデータに対する計算処理手続きの集まりである。トランザクションアプリケーションにおける共通データベースの参照・更新が必要になる。



例えば、予約業務に適用した例を図示すると、下記のようなシステムが挙げられる。



用者が平均的に満足のいくようにしかけを行っている。

- ・ ターンアラウンドタイムの保証
- ・ スループット向上維持

このため、投入されたジョブをコンピュータの使用量に応じて分類している。これには、ジョブクラスという概念を導入して、ジョブクラス毎にジョブの待ち行列を作っている。ジョブクラスは、CPUの使用量、主メモリの使用量、特殊な装置の使用状況によって分類される。これにより、ターンアラウンドの保証、スループットの維持が行われシステムが安定的に運用されることになる。

スループットの向上の障害になりうるのは、ジョブを読むためのカード入出力装置、ジョブの結果を出力するための印刷装置のような比較的低速な機器からの制約である。これを少なくするために、カードリーダーからの入力ストリームが、高速なディスクに束ねて格納され、OSからのジョブの入力は実態的にはディスクから行うことにより高速化される。また、プリント出力結果についてもOSは直接プリンタに出すのではなく、ディスクにはきだしをしておいて、別途ディスクからプリンタへ出力されることにより全体の効率は上がる。

このようにOSは低速の入出力機器からジョブを読んだり結果を出力したりするのではなく、一次記憶装置を使って、オフライン的にカードリーダーから高速のディスク装置へ、また同様にディスク装置からプリンタへ出すことによりシステム性能を上げることができる。

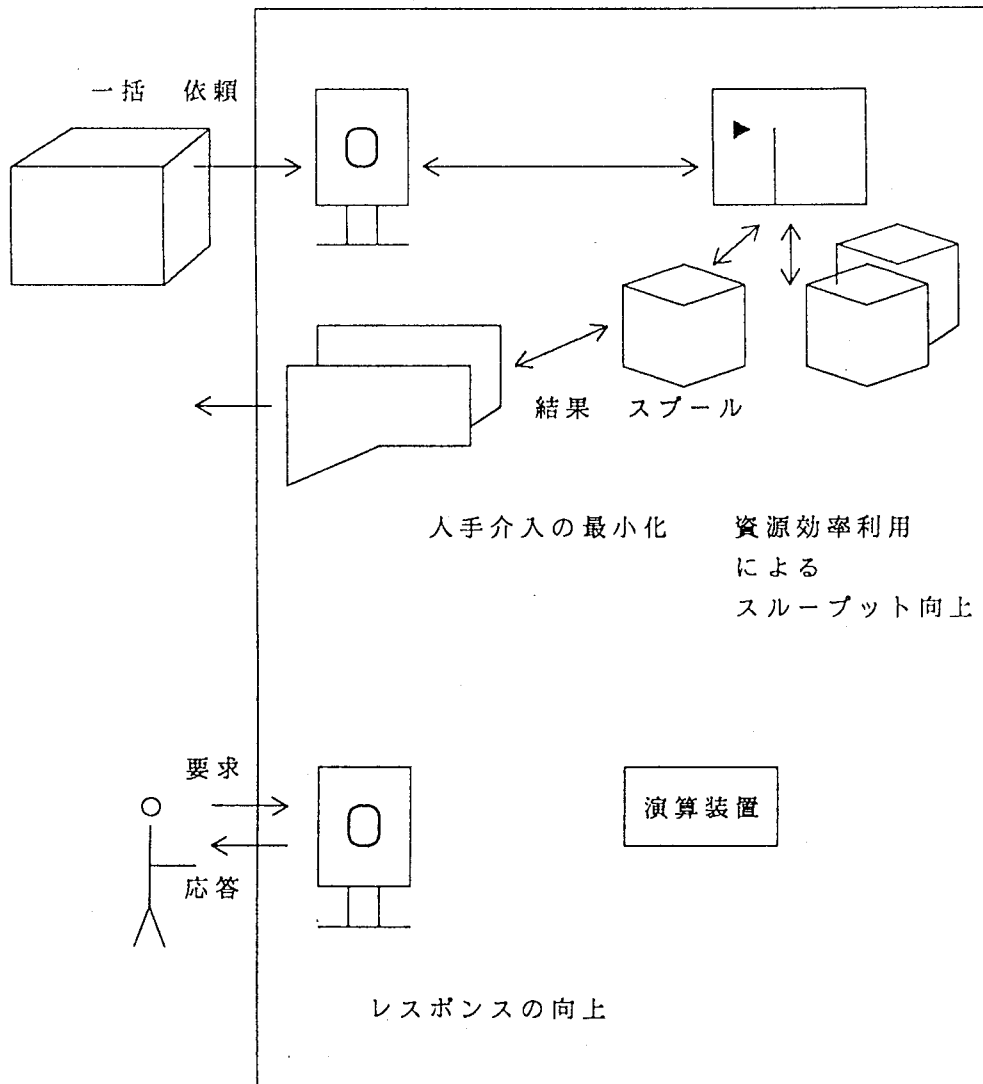
このメカニズムをスプーリング (SPOOL: simultaneous peripheral operations online) という。このスプーリングの存在については利用者のプログラムは知らなくてよい。プログラムはカードリーダーからデータを読んだり、プリンタへ出力していることをイメージしながら入出力をしていけばよく、スプールの存在は意識にはない。カードリーダーからの読み込み要求があるとOSの内部でそれがディスクスプーラからの読み込みに変えられ、また、プリンタへの出力要求があれば、やはりOSの内部でディスク装置にはきだすように取り扱われている。

③ 他のオペレーティングシステムとの比較

1960年代にはバッチ処理の中で即時性の高い処理要求が出始め、オンラインリアルタイム処理のオペレーティングシステムが開発された。これをオンライントランザクション処理と呼んでいる。いまここで、バッチ、対話型、オンライントランザクションの処理の特徴をまとめておこう。

	バ ッ チ 処 理	対 話 型 処 理	ト ラ ン ザ ク シ ョ ン 処 理
システム資源の効率	資源の利用率高い	高 く な い	高 く な い
ス ル ー プ ッ ト	高 い	高 く な い	高 く な い
レ ス ポ ン ス	長 い	短 い	短 い

図表Ⅱ-9 バッチ処理との比較



(2) ジョブ制御

① ジョブ制御言語

- ジョブの実行に関連する情報をオペレーティングシステムに指示するための言語
- ア. ジョブステップやデータの区切りを指示する。
 - イ. ハードウェア資源の割当てを指示する。
 - ウ. 実行するプログラムやライブラリを指示する。

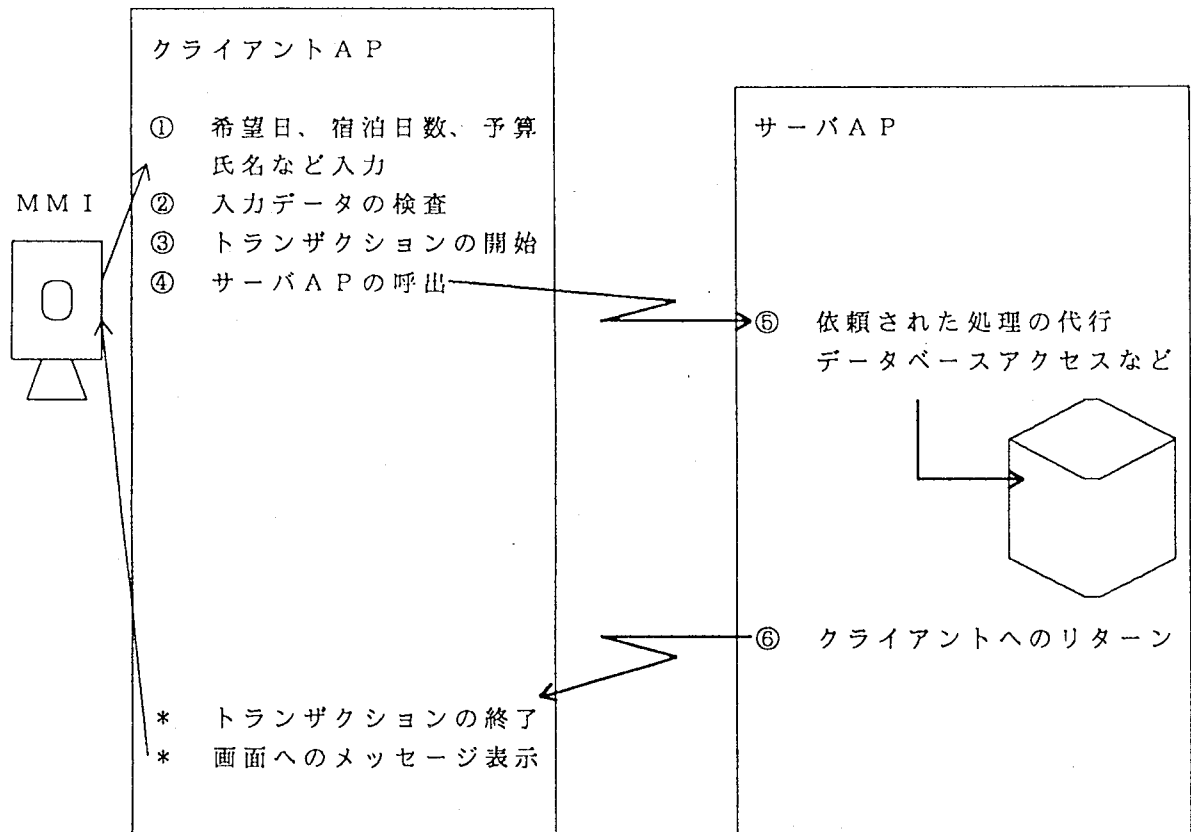
② ジョブの構成

1個以上の順序づけをされたジョブステップ（実行するプログラム名で）を指示する形式をとる。各ジョブステップ毎に実行するプログラム名の指示、利用したいライブラリ名の指示、ファイルの確保・削除の指示、データ入力の指示などを行う。

///JOBNAME	翻訳、編集、実行するジョブ
///STEP1 EXEC PGM =	コンパイルを指示
///STEP2 EXEC PGM =	リンクエディットを指示

⑤ クライアントサーバ型のトランザクション処理

図表Ⅲ-6 クライアントサーバ型のトランザクション処理例

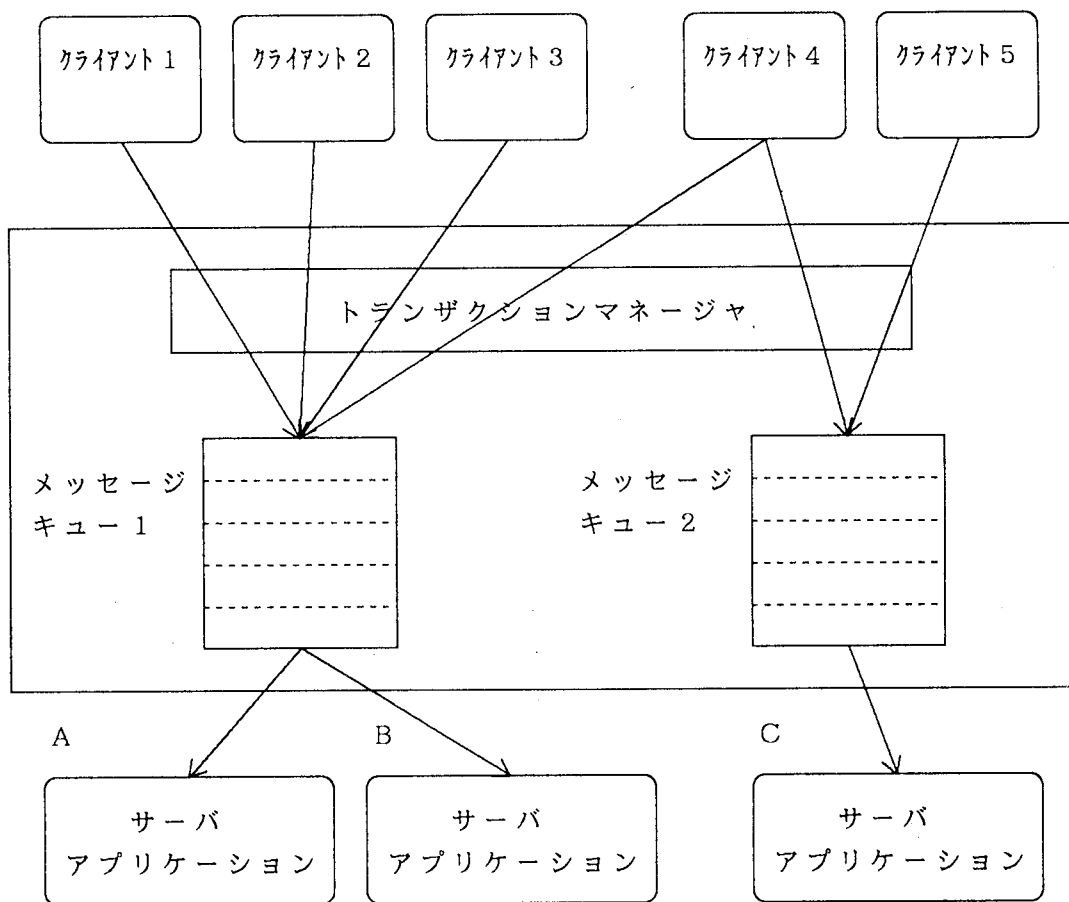


以下で、クライアントサーバ型トランザクション処理の流れを追う。まず、クライアントの AP は端末から MMI を介して、受注情報を入力する。当該入力に関する正当性をチェックし、必要があれば加工をする。

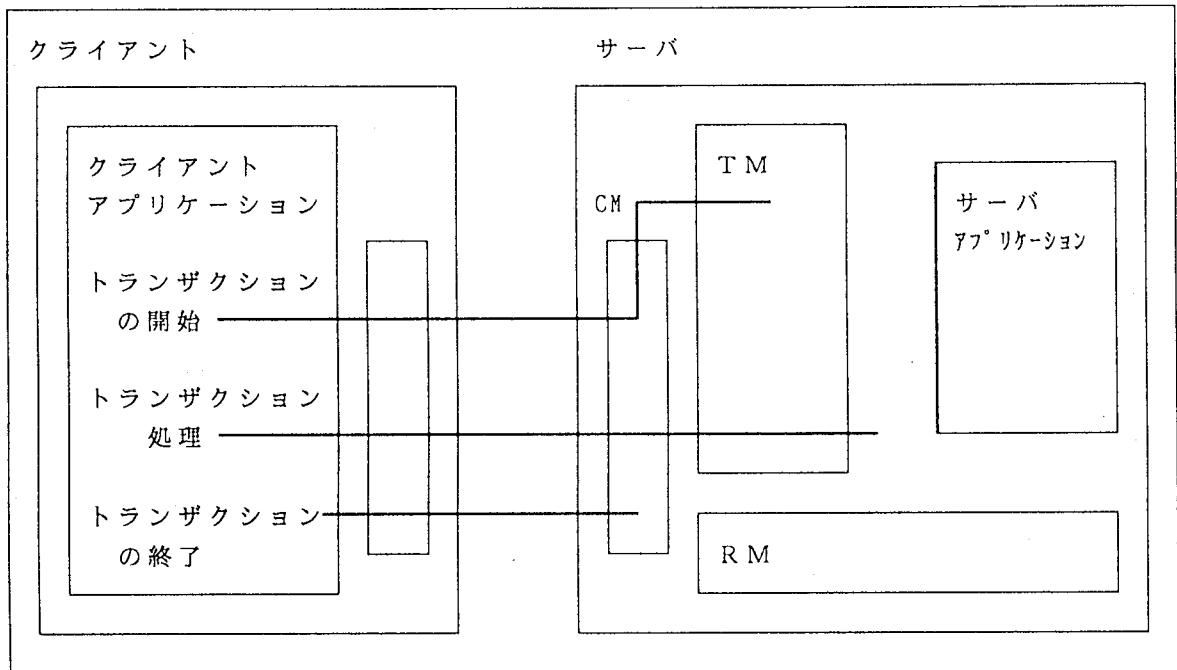
- ・ その後、クライアントは、トランザクション処理の開始指示をする。サーバ側にその旨が送信される。
- ・ サーバ側は、CM を介して TM が受取り、トランザクションを生成し、該当するサーバ AP との対応付けを行う。
- ・ サーバ AP は、トランザクション処理の開始をトランザクション処理モニタに依頼する。
- ・ TM は RM に対してトランザクション処理の開始を通知する。これは XA インタフェースを介して行う。
- ・ 次にクライアント AP はトランザクション処理要求を行う (通常これはデータベースへのアクセスを意味する。)
サーバ側にその旨が送信される。
- ・ サーバ側は、CM を介して TM が受取り、当該サーバ AP に対してトランザクションを振り分ける。
- ・ サーバ AP は、業務処理を進める上で、RM に対してデータベースのアクセス要求を出す。ここで SQL コール形式をとることができる。
- ・ RM はデータベースアクセスを行う。

- ・ 業務が終了すると、サーバAPはTMに対してトランザクションの終了を依頼する。
 - ・ これを受けてTMは、RMに対してトランザクションの終了を指示する。これもXAインタフェースを用いて行う。
 - ・ サーバAPは、業務処理結果についてクライアントへの返送をTMに依頼する。
 - ・ TMは、業務結果をクライアントへCMを介して送る。
 - ・ クライアントAPは、ここでトランザクションの終了であれば、それを通知する。
これはサーバ側に送られ、TMはRMに対して完了（コミット）を指示する。また、トランザクションを消滅する。
- トランザクションの振り分けとは、クライアントから受信したトランザクションを振り分けるためにメッセージキューにつなげる。これを介してサーバAPとの対応が付けられる。

図表Ⅲ-7 トランザクションの振り分け



図表Ⅲ-8 クライアントサーバ型トランザクション処理のソフトウェア



⑥ 分散トランザクション処理に関連する標準化

a. 分散型トランザクション処理の規定

X/Openにおける分散型トランザクション処理の参照モデルが規定されている。

b. XAインタフェースの規定

X/Openの共通アプリケーション環境(CAE)仕様としてTMとRMの内部インタフェースが規定されている。

c. TXインタフェースの規定

X/OpenのCAE仕様としてTMのAPIであるTXインタフェースが規定されている。

d. CMのAPI規定

AP間のインタフェースがCM仕様の規定によりいずれ実現される。

クライアントサーバ型、会話型、RPC(リモートプロシージャコール)型が候補となっている。

ア. XATMI(クライアントサーバ型)

クライアントからサーバ側のサービスを呼び出す方式

イ. Peer-to-Peer(会話型)

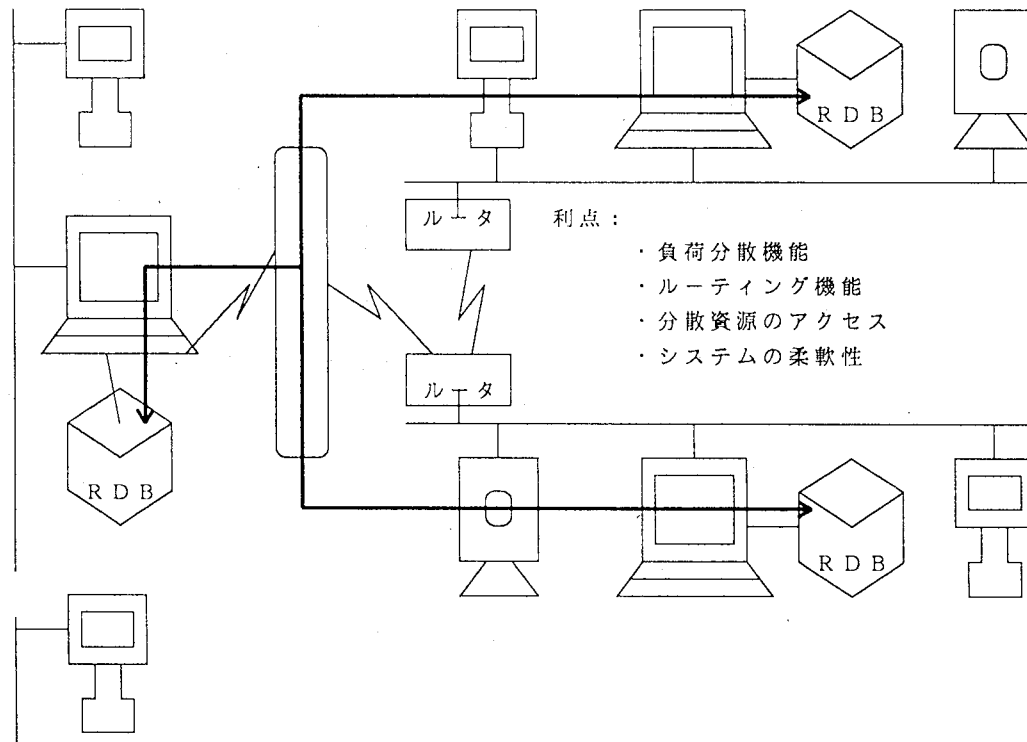
IBMのSNAのトランザクション処理プロトコルLU6.2をベース

ウ. TxRPC(RPC型)

(2) 協調分散機能を持つトランザクション処理モニタ

複数のUNIXマシンサーバで協調分散により柔軟なオンライントランザクション処理システムを構築することができる(図表Ⅲ-9)。サーバ間の通信処理は、各サーバ上のTM間で行われる。

図表Ⅲ-9 トランザクション処理モニタの協調分散機能



① 負荷分散機能

TMが、各サーバの負荷状態を監視しており、パフォーマンスを予想しながら自動的に負荷分散を行うことができる。

② トランザクションのルーティング機能

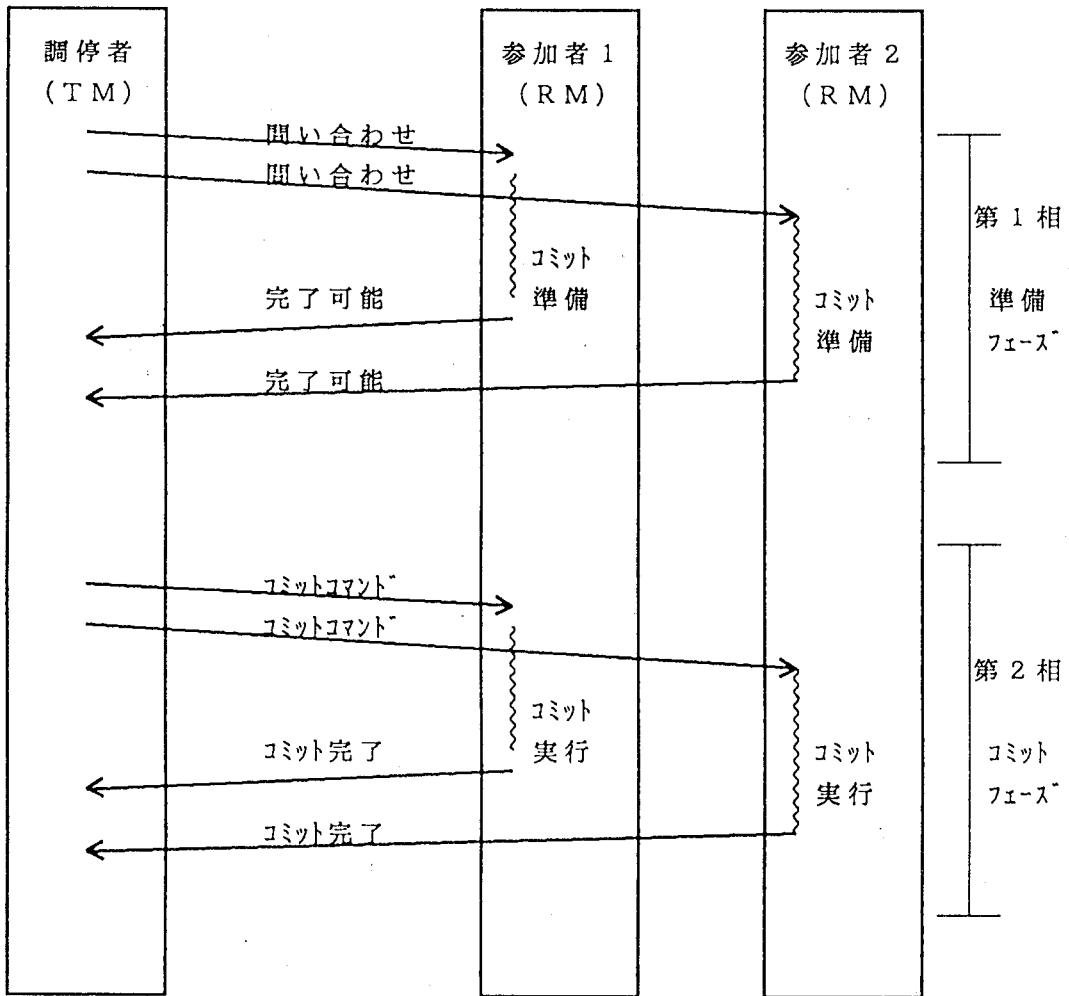
ダウンしたサーバから処理を他のサーバに移す機能および宛先の付いたサーバへルーティングする機能を持つことができる。

③ 分散資源のアクセス

あるクライアントからのトランザクションを複数のサーバ上に分散し、かつ複数のリソースマネージャ (RM) に対してデータアクセスの依頼が可能である。

サーバAPがTMに対してトランザクション処理の終了を要求すると、TMがXAインタフェースが提供する2相 (フェーズ) コミットと呼ばれるコミットメントプロトコルにより、RMに対してトランザクション処理の終了を指示する (図表Ⅲ-10参照)。

図表Ⅲ-10 2相(フェーズ)コミット



3. オンラインデータベース

(1) オンラインデータベース管理

① データ定義機能

外部スキーマ、概念スキーマ、内部スキーマを記述する言語機能を使って、データベースの構造および整合性制約を宣言する。

CODASYLでは、スキーマDDL、サブスキーマDDL、データ記述言語DDLが提供される。

RDBでは、SQL-DDLで提供される。

② データ操作機能

データの基本操作(データ検索、挿入、更新、削除)はオフラインでもオンラインでも同様に行う。

データベースを操作するためには、一つにはデータベースとは独立な汎用言語を想定し、データベースのサービス機能に対応させて基本機能をサブルーチン(あるいはライブラリ)呼出手続きで利用する“親言語方式”と呼ぶインターフェースがあり、COBOL、C、FORTRANなどの既存言語で扱うことが可能となっている。

一方、利用者に特定の言語を提供し、その言語機能の中でデータサービスの機能を提供する方式を自立言語(あるいは利用者言語)と呼ぶ。

③ データ制御機能

a. データ保全機能

b. データ機密保護機能

データの種類によって、アクセス禁止、参照のみ許可、参照・更新とも許可のモードを設定し、どの利用者にどこまでの権利を与えるかをデータ管理者が管理を行う。

これにより、アクセス権限のないものからの不正使用からデータベースを保護する。

データ機密保護を行うために、ユーザIDおよびパスワードの組み合わせで対策を講じる場合が多い。これは、DBMSの保護機能とOSの持つ保護機能を組み合わせて実現される。

c. その他の制御機能

複数利用者による同時期の同一データへの更新に際して、データアクセスに矛盾を来さないよう適切な制御が必要である。これを同時実行制御あるいは排他制御と呼ぶが、このメカニズムは、オペレーティングシステムの共有資源へのアクセスの制御と同様な考え方で実現できる。

この同期制御を実現するメカニズムには各種の方法が存在する。ロック方式、時刻印方式、楽観的方式などがその代表例である。

ア. ロック方式

あるプログラムがデータをアクセスしているときに、他のプログラムが同一データをアクセスしようとしたときにそのプログラムを待たせる方法である。ロックにも厳密には、

- ・ 排他ロック（他のトランザクションからのいかなるアクセスも排除する。）
- ・ 共有ロック（競合するトランザクションでも同時読み取りは可能とする。）

とに分けることができる。

ロックをかける情報の大きさは、レコードのフィールド、レコード全体、データベース全体などレベルが考えられる。どれを採用するかは、データベースアクセスの全体的な効率にどう影響があるかによって決めるべきものである。

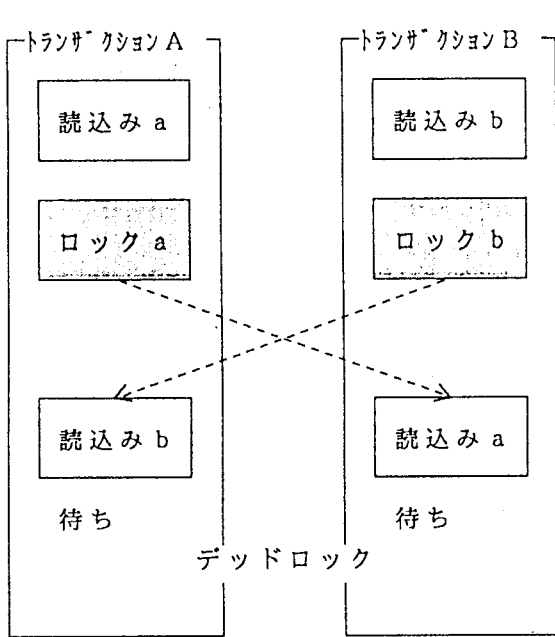
ロックの単位が大きければ、より完全なロックに近づくためロックを実現する処理が簡単であり、また、信頼度も向上する。反面、トランザクションの待ち時間が長くなったり回数が増えたりして全体的な効率が落ちる。

ロックの単位が小さければ、トランザクションの待ちは少なくなり効率は上がるものの、頻繁なロックとその解除によるオーバーヘッドが大きくなり、信頼度も低下する。

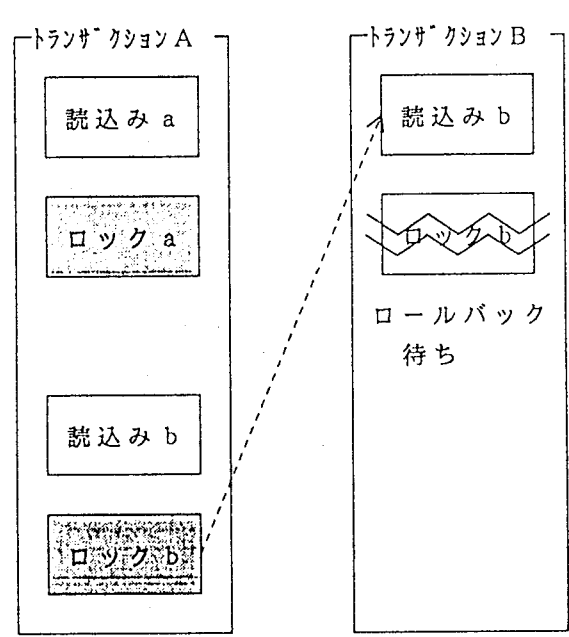
排他制御によるロック方式では、デッドロックの発生に注意する必要がある。この状態は、二つのトランザクションが互いに既にロックを掛けているデータに対して、それぞれがさらにロックを掛けると互いの永久的な待ちにおちいる現象をいう。

ここで、DBMSが一方のトランザクションを開始直前の状態にもどし（これをロールバックと呼ぶ。）、トランザクションを最初からやり直すことによりデッドロックを解除できる。

図表Ⅲ-11 デッドロック



図表Ⅲ-12 デッドロック解除



イ. 時刻印方式

トランザクションの開始時刻やデータの読み込み、書き出し時間を記録し、それらを内容比較してデータの更新順序を決める方式である。この方式では、デッドロックは発生しないが、分散データベースのような複数サイトでの順序決めを行うには技術的な課題も多い。

ウ. 楽観的方式

データのコピーをとり、各々のトランザクションが更新結果を反映させる時点で同一データへの更新に重複がないかどうかをチェックする。

(2) マルチスレッド

クライアントサーバ・アーキテクチャにおいて今後普及の期待が大きい技術としてマルチスレッドがある。スレッドは、オペレーティングシステムからみたスケジューリングの最小単位であるプロセスの中に、さらに小さな実行単位として位置づけられる制御の単位である。スレッドは、オペレーティングシステムが直接スケジューリングの対象とする訳ではないので、プロセスのような属性（プロセスの制御テーブルなどに記述される。）を持つものではない。

スレッドは、実行対象の切換をプロセスの属性情報の退避と回復を伴うようなプロセススイッチングによって行うのではなく、もっと高速に切り換えられる。

スレッドを導入する目的は、サーバが数多くのクライアントからのサービス要求に対して、個々のクライアントに対応してプロセスを割り当てる方式をとると、OSコールにともなうオーバーヘッドが大きくなり効率が低くなることを回避したいためである。

クライアントの要求をスレッドとして同時に複数の実行と制御を行うことになる。これをマルチスレッド技術と呼んでいるが、従来のマルチプロセッシングよりも少ないオーバーヘッドで並列的な処理が可能であり、商用のDBMSでの採用が増えていく方向にある。

(3) 分散データベース

ネットワークコンピューティングにおいては、自マシンにないデータベースであってもデータベースの所在を意識しないでアクセスすることができる。ネットワークには、あたかも一元管理

されたかのような仮想的なデータベースを構成でき、これを分散データベースと呼んでいる。

利用者が物理的な分散状態を意識せずにデータを相互利用できる。すなわち、自マシン、他マシンにかかわらず透過的にデータの参照、更新ができる。

このシステムを構築するためには、以下のような仕掛けが必要となる。

- ① データディレクトリ機能
データの所在情報を一括管理する。
- ② クライアントサーバ機能
アプリケーションプログラムからのデータ操作要求に対して、クライアントマシンに処理依頼をさせ、サーバマシンに実際の操作の代行と結果の返送を行わせる。
- ③ トランザクション制御機能
一つのトランザクション処理で複数のデータベースを更新する場合、同期制御が必要となる。
- ④ データの整合性の維持
自マシン他マシンにかかわらず全体を一つのデータベースとするために、全体的なデータの一貫性の維持が必要
分散データベースシステムの同時実行制御では、単純にロックをかければ済むというものではない。複数サイトのデータに対してロック待ちが多発しかねず、集中型のデータベースでのロック制御よりは複雑である。
 - ・ 分散同時実行制御 ロック待ちの状態監視、デッドロック検出時のロールバック対象の選択など。
 - ・ 分散トランザクション制御 コミットメント制御機能
 - ・ データ辞書の分散化

(4) バックアップとリカバリ

データベースシステムの運用中にソフトウェア障害あるいはハードウェア障害によって、データベースが論理的なあるいは物理的な不完全状態を生じる恐れがある。それに対処するバックアップと、障害発生に対する回復処理機能が必要になる。

- ① バックアップ機能
 - a. バックアップ内容
 - ア. データベースの完全なバックアップコピー
一貫性の保証されたデータベースの別媒体へのコピー
 - イ. 差分情報の完全なダンプ
 - ウ. 完全な障害復旧のためのログファイルの保存
データベースへの更新履歴。更新前ログ、更新後ログ、トランザクションコードの情報など。
 - エ. チェックポイント情報の保存
- ② リカバリ処理
 - ・ ロールフォワード
コピーされたバックアップデータを記憶媒体上に復元、バックアップ時からのログをデータベースに適用
 - ・ ロールバック
最新のチェックポイントから逆方向に見て行ってデータを元に戻す。
 - ・ 指定したチェックポイントまでのリカバリ
 - ・ 他の業務（OLTPなど）との連携によるリカバリ

4. プログラムの構造

オンライントランザクション処理では、同じ業務アプリケーションプログラムに対する複数のプロセスからの同時処理要求をサポートする必要がある。また、その要求は即時に処理しなければならず、シリアルに処理しては対応が遅れることもあり、パラレル処理を可能とする処理手続きが必要であり、これをリエントラントプロシージャによって実現することができる。リエントラントプロシージャの特徴は以下の通り。

① マルチタスク

同時発生するトランザクション処理をパラレルに処理するために、各トランザクションをプロセスとして動作させる必要がある。リエントラントプロシージャであれば各トランザクションに固有な情報を別々に持つことができるので、同じプログラムを並列動作できる。

② プログラムとデータ領域の分離

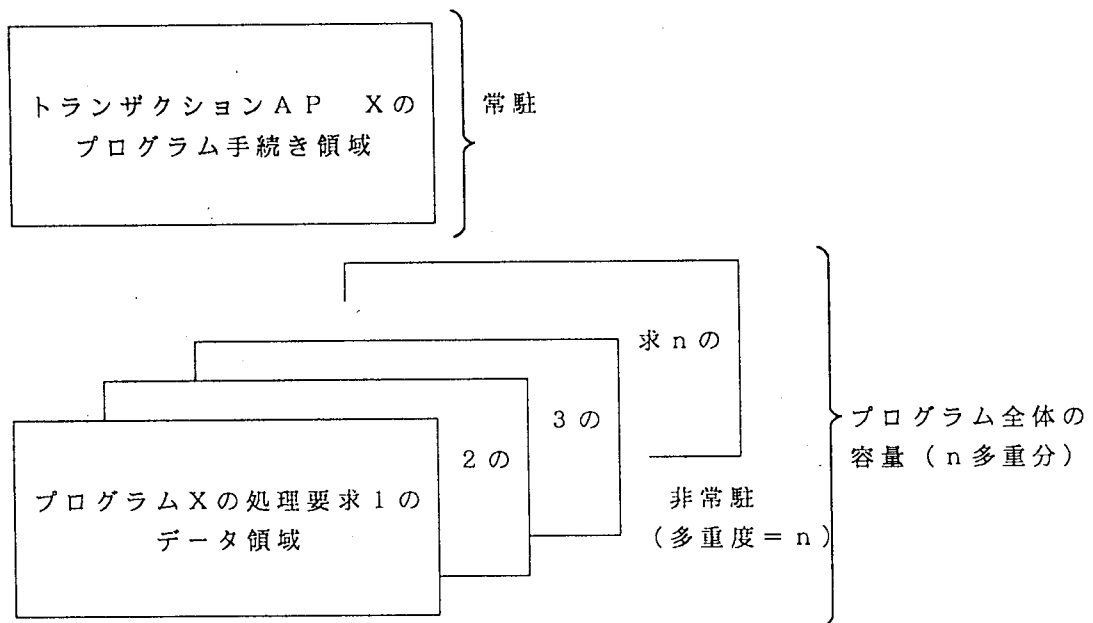
同一のプログラムを同時に異なるプロセスがパラレルに動かすためには、プログラムの処理手続き部とデータ部とが分離されている必要がある。これが分かれていないと、あるトランザクションを処理を中断して他のトランザクションを処理しようとするときデータ部の内容が変わってしまうため、トランザクション処理の再開がきかなくなる。

このため、データ部はトランザクションが発生するたびに個々に領域を獲得して割り当てる必要がある。これを行わなければリエントラントプロシージャにはならない。

③ メモリの節約

リエントラントプロシージャを使えば、トランザクション毎にプログラムをメモリ上に用意する必要はなくなり、空間を節約することができる（図表Ⅲ-13参照）。

図表Ⅲ-13 リエントラントプロシージャのメモリマップ



プログラムの手続き領域は一つだけあればよい。データ領域は手続き1個について予めn個分用意しておくか、あるいは要求発生時に領域を確保して処理が終了したら解放する方式をとる。

5. 分散型トランザクション処理の利点

大規模中央集中処理型のトランザクション処理では、情報共有が容易で、全ての端末から簡単にアクセスでき、また、端末間のコミュニケーションが不要である。そのうえ、運用・管理も容易である利点がある。

しかしながら、オンライントランザクション処理システムの構成要素が緊密に結合されていると、次第に不利な点が生じる。例えば、新規のアプリケーションシステムの追加が非常に複雑でコスト高となる。これは、新規のアプリケーションが既存のサービスに影響がないことを確認するために、モニターとその機能全てを再試験しなければならないからである。また、その試験のために全システムを止める必要が出てくる。全てのシステムの要素が緊密に結合されていることに原因する。

さらに、単一のTMではフォールトトレランシーとアベイラビリティに大きな問題が出てきた。構成要素のどれか [データベースの処理論理、通信処理、アプリケーションプログラムなど] が故障した場合、システム全体に影響する可能性が強い。これは、全ての要素が共有メモリ情報にアクセスするからである。このような環境では、あるモジュールの障害に際し、他の要素が使っている共有メモリーが簡単に壊されてしまうからである。

この問題は既に分散型トランザクション処理の導入により解決できると評価されている。

分散型トランザクション処理では、トランザクション作業を小さな論理ブロックに細分化できる。細分化された要素は自律的なものとなる。

分散型トランザクション処理の利点としては、

- ・ 小さな独立したコンポーネントとなっているのでシステムの成長と変化に適応が容易である
- ・ 作業がそれぞれ専門化されたコンポーネントにより行われるため、他のアプリケーションに悪影響を与えないことを確認しなくても新しい機能を追加できる
- ・ オンライントランザクションシステム全体を止めずに済むためアベイラビリティも向上する
- ・ 対故障性が強いこともあり、アベイラビリティも向上する
- ・ 一つのアプリケーションまたは通信に障害が発生しても、全体環境をダウンさせることはない
- ・ 分散型トランザクション処理はオンライントランザクション作業をベストな拠点で実行できるが挙げられる。

6. 主要用語

オンライントランザクション処理	ACID 特性	原子性
一貫性	分離性	持続性
クライアント	サーバ	応答時間
ターンアラウンドタイム	フォールトトレラントマシン	
リカバリ	ログ	バックアップ
データベース	GUI	DBMS
TPC	TPC - A	TPC - C
クライアントサーバ型	分散型トランザクション処理	
TM	RM	CM
コミット	XA インタフェース	TX インタフェース
XATMI	Peer - to - Peer	TXRPC
コミットメントプロトコル	2 相コミット	CODASYL
関係型 DB	親言語方式	データ保全機能
データ機密保護機能	同時実行制御	ロック
デッドロック	障害回復機能	マルチスレッド
データの一貫性	分散データベースシステム	分散同時実行制御
分散トランザクション制御	データ辞書の分散化	バックアップ
リエントラントプロシージャ	マルチタスク	プログラム域
データ域		