

第VI章 分散処理システム

第Ⅵ章 分散処理システム

学習目標

1. 処理、データ、管理のそれぞれについて、分散することの得失を理解させる。
2. 分散処理システムの長所、短所を正確に理解させる。
3. オープンシステムとすることの問題点を理解させる。
4. 分散処理システムの「透過性」と「分離」、その重要性について理解させる。
5. ハードウェアの水平分散構成と垂直分散構成を応用プログラムの構造と関連させて理解させる。
6. 分散処理応用プログラムが動く基盤となるプラットフォームについて理解させる。
7. 分散処理機能を持つオペレーティングシステムを分類し理解させる。

全体概要

計算機処理は、1970年代中葉までの汎用コンピュータによるバッチ処理主体の計算処理と入出力全盛の時代から、汎用コンピュータ上でTSS形態による対話的なコンピュータ利用が可能となることにより次第に対話型処理が幅をきかせ始めた。1980年代に入りミニコンピュータの急速な普及、パソコンの導入、1980年代半ばにはワークステーションの利用へと対話型のコンピュータ利用が圧倒的に増えていった。

近年、バッチ処理は影に隠れながら相変わらず根強く生き続けてはいるものの、コンピュータ利用者の間では私語に近い状態にある。本章では、ことさら対話型処理という言葉を強調するのではなく、むしろ現在のコンピュータの自然な使い方は対話が基本となるという観点でその特徴を解説する。

内容のあらまし

節 項	内 容
1. 分散処理 (1) 分散処理システム (2) 分散の対象 (3) 透過性と分離 (4) 分散処理システムの特徴	集中処理システム、分散処理システム 負荷、処理、データ、管理の分散 透過性、分離 長所、短所、異機種による分散処理 システムの利点と問題点
2. 分散処理システムの構成 (1) 垂直型分散処理システム (2) 水平型分散処理システム (3) マルチベンダの分散情報システム事例	
3. 分散処理システムにおける処理の構造 (1) 分散処理システムのソフトウェア構造 (2) アプリケーションのプラットフォーム (3) クライアントサーバ型処理 (4) ピアツーピア型処理	応用プログラム、オペレーティングシステム 通信プロトコルソフトウェア 仮想ファイル、分散データベース プリンタの共用、応用プログラムの共有 管理情報の共有、CPUの共有と負荷分散 ピアツーピア型処理ネットワークOSの定義
4. 分散処理オペレーティングシステム (1) 分散資源管理 (2) ネットワーク機能を持つオペレーティングシステム (3) ネットワークオペレーティングシステム(NOS)	名前の管理 UNIXのネットワーク機能、NFS NIS 基本構成 ネットワークOSの基本機能 ネットワークOSの代表的なサービス
5. 主要用語	

1. 分散処理

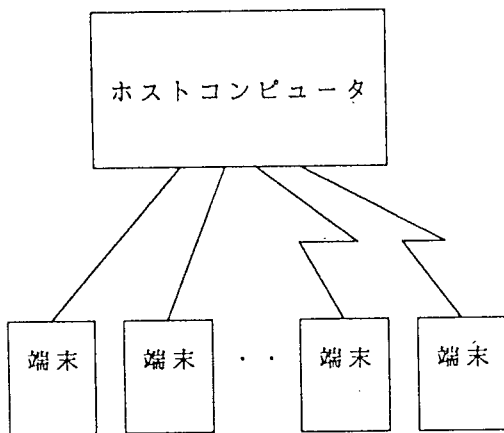
(1) 分散処理システム

分散処理は、集中処理とは異なる情報処理構成を指す。汎用メインフレームコンピュータに全体を制御される情報処理形態の中に、1960年代の後半頃からミニコンピュータが登場して以来、分散処理という言葉が使われ始めた。今まで全てのコンピュータ処理はメインフレームコンピュータで行ってきたことが、小さな仕事はミニコンピュータでこなそうという発想が定着し始めた。この時点では、必ずしもメインフレームとミニコンピュータは物理的にはつながっていなかった。

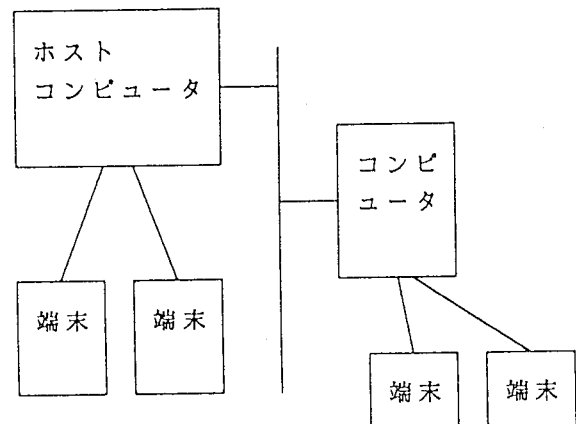
大型の計算処理でなければ、ジョブを依頼し結果を受領するまでの時間的なオーバーヘッドの大きい計算センターマシンを使う必要はない。コンピュータ使用料も安くできる。特定の業務部署に導入したミニコンピュータで処理プログラムの開発者自らがオペレーションも行って計算を行うわけだ。これが分散処理の始まりである。

1970年代半ばにもなると、LANが登場しメインフレームとミニコンピュータがつながるようになりユーザ端末から必要に応じて手元のミニコンピュータを使うか、メインフレームコンピュータを使うかをその都度判断して切り分けられるようになった。やがてパソコンの登場により、マイクロメインフレームリンクという言葉も使われはじめ、スケールの異なるコンピュータが互いにそれぞれの役割を与えられて全体的に統括されるようになってくる。この頃になると、色々な形態の分散処理システムが構成されるようになった。

図表VI-1 集中処理システム



図表VI-2 分散処理システム



分散処理システムでは、それぞれのコンピュータは同じメーカーの同じ製品である必要はない。他のメーカーの製品でその上異なるオペレーティングシステムを持つコンピュータであってもかまわない。コンピュータ間のリンクは、物理的には通信制御の部分、論理的にはOSの一部あるいはアプリケーションによって実現される。

それぞれのコンピュータでどのような仕事を行うかはシステムの導入者が計画的に検討すべき事柄である。計算負荷をどこに掛けるか、対話による入出力負荷をどこに掛けるか、データベースをどこに置くかなど全体のバランスの中で決定される。

(2) 分散の対象

① 負荷の分散

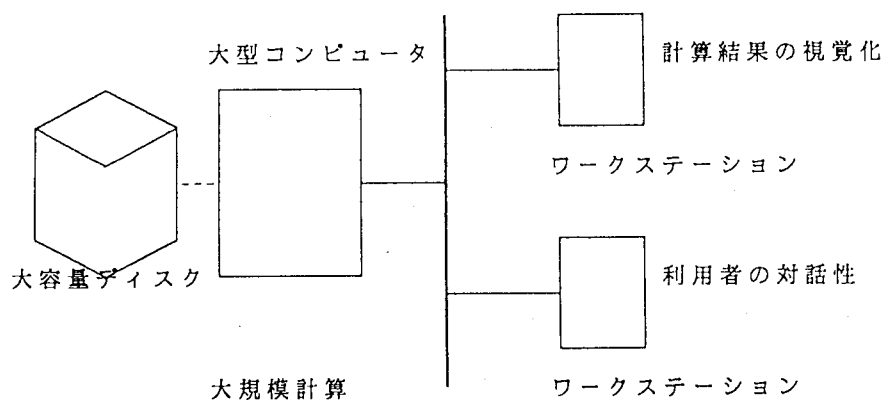
予想される計算の量からみて、どの程度の処理性能を持つコンピュータに計算を行わせる

か、大量の技術計算を高速に解きたいのか、多少時間がかかってもよいから手元のコンピュータで処理したいのかなどにより、割り振られるコンピュータが決定される。

集中処理全盛時代の末期には利用者の対話による負荷がメインフレームコンピュータにのしかかり、ユーザインタフェース処理は分散コンピュータで行うよう急速にオフロードされていった。

大量の計算結果をディスプレイ上に表現することをメインフレームで行うには負荷が多すぎ、メインフレームの意義が小さくなる。その後も分散コンピュータにどんどん移っていった。

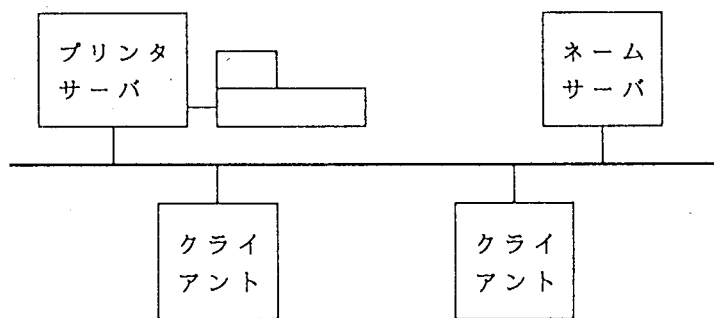
図表VI-3 負荷の分散



② 処理機能の分散

大量のプリント処理、ネームに関連する処理などは自コンピュータで行うことはCPUの稼働率や利用者の環境を低下させることにもなり、特定のマシンで専門で行うことが多い。プリンタサーバ、ネームサーバなどと呼ばれるコンピュータで特定の機能に対し複数の利用者からの要求に応える。プリンタサーバには単にデータを印刷出力するだけでなく、出力の頁毎に頁番号を付加したり、印刷用の文字体系に変換したりする機能を置くこともできる。

図表VI-4 処理機能の分散

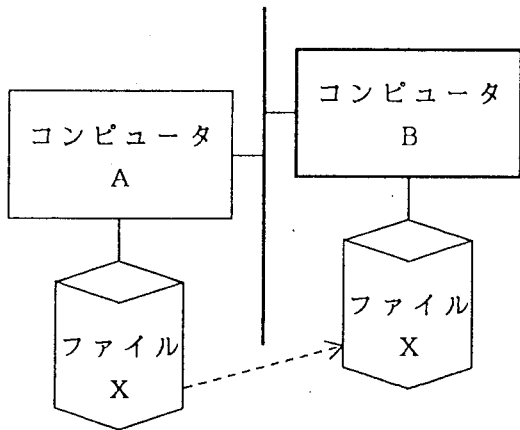


③ データの分散

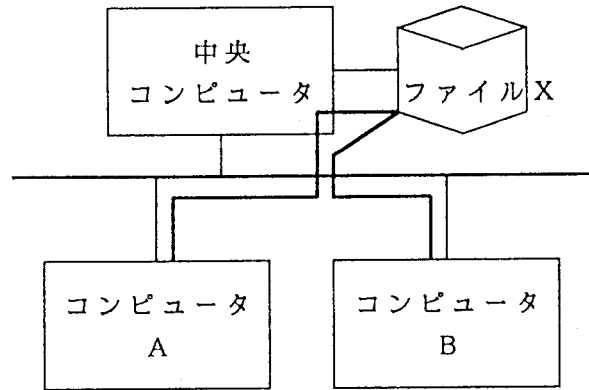
それぞれのコンピュータが頻繁にアクセスするようなデータファイルはそれぞれが手元に置く。これによりアクセス時間が向上するとともに、他のコンピュータの停止時にアクセスが不可能になるリスクが回避できる。

共用データファイルの更新は一カ所のコンピュータで集中的に行いセキュリティを確保する。また、常に中央コンピュータのデータを更新する方式をとらず、手元のコンピュータで更新をし、ある程度蓄積された段階（週次、月次など）で中央に送るような分散も考えられる。

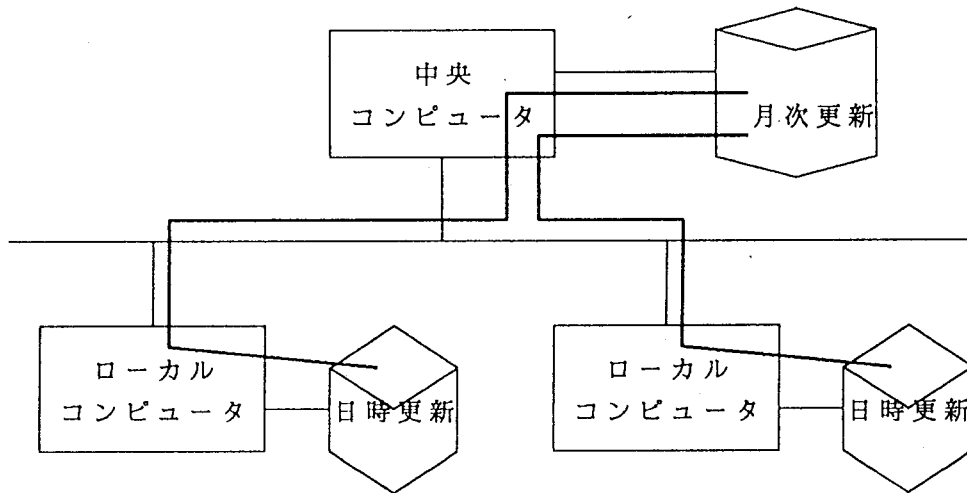
図表VI-5 データの複製



図表VI-6 中央で集中更新（非分散）



図表VI-7 分散かつ一括更新

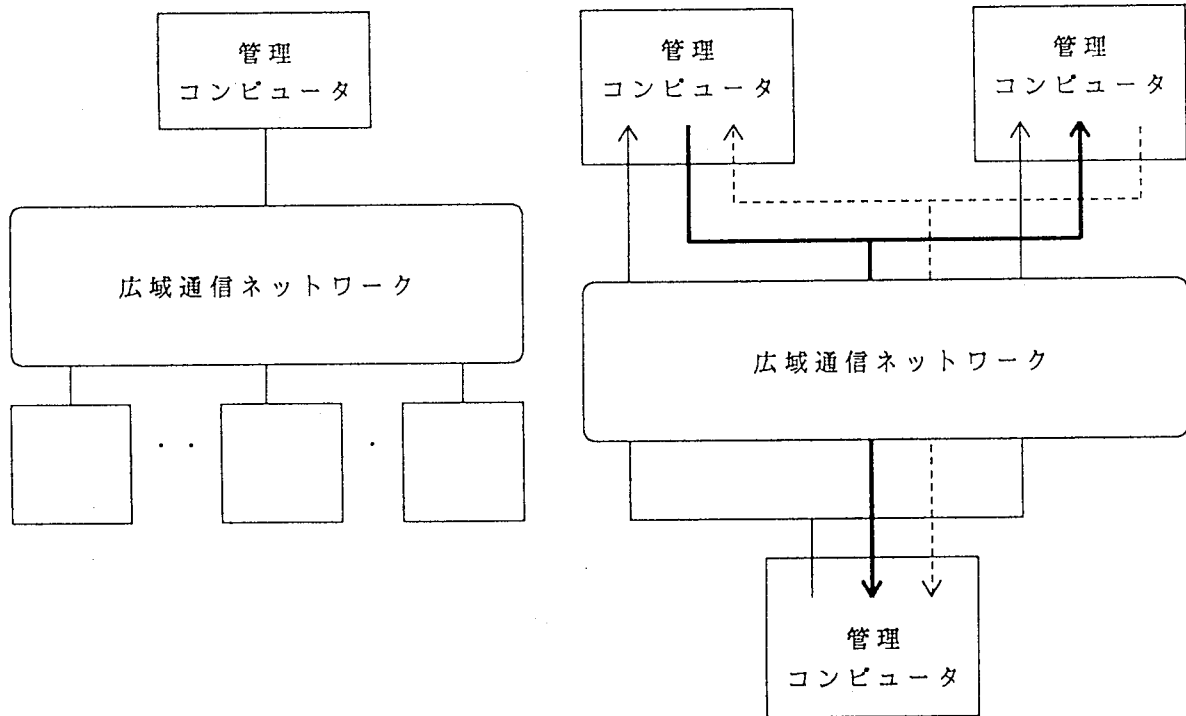


④ 管理の分散

ネットワークでつながっている全てのコンピュータリソースを各々のコンピュータから偏りなく、全体的なバランスがとれ、応答性の落ちない資源の使い方が必要である。このような効率の高いシステム運用を行うためには、ネットワーク上の各資源の使用状況（負荷、量など）コンピュータ間の通信量が把握されなければならない。

そのために資源に関する管理情報が必要となるが、システムが広域に亘り規模が大きくなると管理コンピュータの負荷が増大化し、逆に効率運用を妨げるものとなろう。これを回避するために、管理情報と管理そのものを分散させる構成を取り得る。

図表VI-8 管理の分散



(3) 透過性と分離

ネットワーク接続された各種分散された資源を利用者が使うときに、物理的な所在を意識せずに扱える仕掛を提供することを透過性 (Transparency) を与えるという。

① 透過性 (Transparency)

a. 位置的透過性

資源の物理的な所在地をプログラムやコマンドで意識せずに使えること。

b. アクセス透過性

自コンピュータの資源に対する操作と他コンピュータの資源に対する操作を全く同じようにできること。

c. 規模透過性

システムやアプリケーションの規模を変更しても、システム構成やプログラムの処理論理に影響を与えないこと。

d. 複製透過性

プログラムやコマンドを使用して、特別な配慮をせずに複製ができること。

② 分離 (Separation)

利用者やアプリケーションプログラムからは自コンピュータの資源のように見えても、実際には別々の資源が割り振られていることを分離という。この場合、例えば、あるユーザの誤操作によるエラー発生があっても、他のユーザには波及しない。

(4) 分散処理システムの特徴

① 長所

a. 一部の資源に障害が発生したり、停止したりしていても、全体には影響を与えない。

b. 機能やデータを自コンピュータに置くことにより、通信の量が減少しコスト低減や性能低

下の抑制をできる。

- c. 大型機での集中管理ではバックアップに膨大なコストを掛ける必要があるが、分散コンピュータではその費用は軽くて済む。
- d. 分散によりコンピュータの機能が独立性を高めるため、それぞれのコンピュータソフトウェアの保守が他のコンピュータソフトウェアに影響を与えない。
- e. データは各分散コンピュータから入力され、入力時点でデータ内容のチェック、入力手作業が行われるため、中央コンピュータの人手の介在が減る。これは信頼性の向上にもつながる。

② 短所

- a. 分散化するための根拠（経済性、性能向上、信頼性、生産性など）を明らかにしなければならない。これには計画性と経験が必要であるが、容易ではない。
- b. 集中戦略を取るべきか、分散戦略を取るべきか、システムの構成について、ハードウェア、ソフトウェアだけでなく、キャパシティプランニング、ソフトウェアの効果的な利用などに関するかなりのスキルが必要となる。
- c. マルチベンダ方式を採ることとなろうが、インタフェースの問題で障害が発生し易く、標準化の技術や、トラブルの切り分け技術に関連して相当高いものが要求される。
- d. 分散コンピュータにおいても運用・保守をおろそかにはできない。利用者だけでは解決できない問題がどんどん発生し得るため、専門家を育成しなければならない。
コスト減を期待し過ぎることは危険である。
- e. 分散コンピュータ周辺では、中央コンピュータに比較するとセキュリティ対策は手薄である。

③ 異機種による分散処理システムの利点と問題点

異機種によるシステムの利点と問題点は裏腹の関係にある。

分散化を進めることは、とりもなおさず利用者に最適な（あるいは好む）コンピュータ資源を選定できることである。また、利用者がそれぞれ優良なシステムを作れば他の人もそれを利用したくなり、相互運用性の要求も出てくる。情報の形式や管理が統合化されれば、その共有も可能となり、業務遂行効率も一段と高くなる。利用者は常に最新の技術を安いコストで手に入れることができることは素晴らしいことである。

しかしながらここに大きな落とし穴が存在する。

まず、異機種（ハードもソフトウェアもデータも全てを含め）によるシステム構成を採用する場合、標準化が確立されている技術であるかどうか、さらに相互接続の実績があるかを確認しておかなければならない。導入者にとっては既存のものを継続利用する必要があり、既存資源と新規導入資源との間でインタフェースが採れなくてはならない。

導入した技術が相当世の中で普及していなければ安心できない。限りなくリスクが予想されてくる。

ここで、異機種環境を作るときに留意すべきことをあげておく。

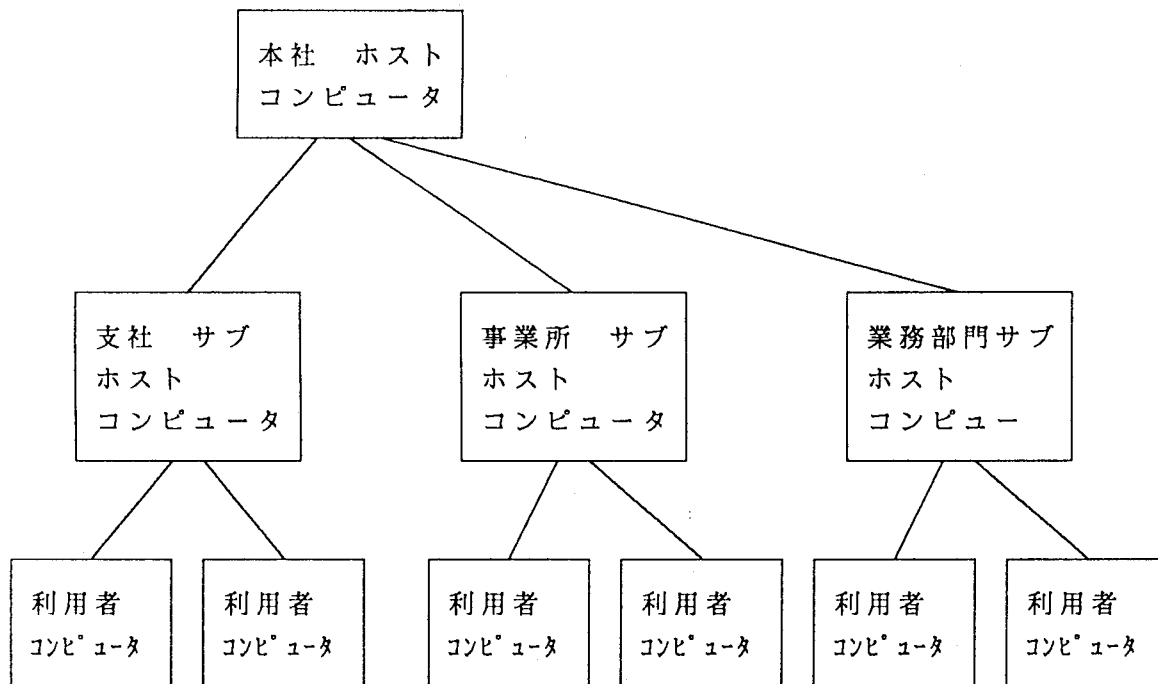
- ・ 導入する資源に関する標準への準拠性を注意深く確認すること。
- ・ 異機種間接続を可能とするハードウェアやソフトウェアが出回っているか、あるいは自力で開発できること（ミドルウェアの存在など）。
- ・ バージョンアップ後に障害が発生した場合、原因の特定や障害対応をベンダに義務づけられること。
- ・ 価格が安いだけでなく、使いやすい、運用・保守にかかる費用が少ない、保証が十分であること。

2. 分散処理システムの構成

(1) 垂直型分散処理システム

階層的に処理が分散されるシステム。企業情報システムであれば、本社にある基幹コンピュータの下に、各支社、事業所、業務部門レベルの分散コンピュータが接続され、さらにそれぞれの分散コンピュータに制御されて専用端末やパソコンがつながっている。

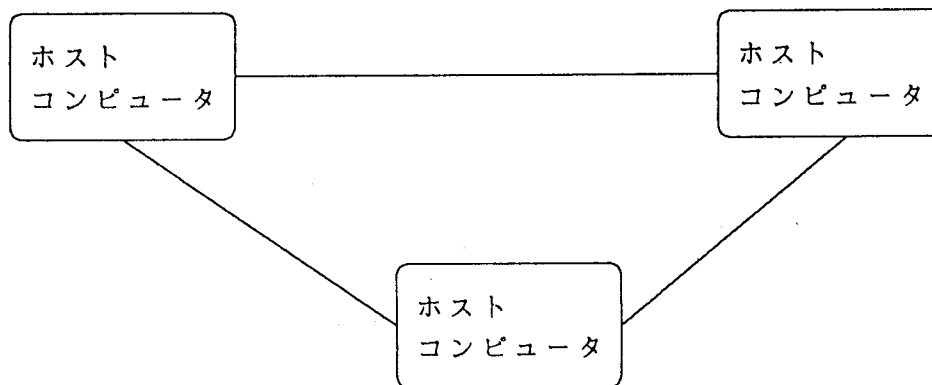
図表VI-9 垂直型分散処理システムの例



(2) 水平型分散処理システム

階層型のシステムには主従関係が出てくる。これに対して、水平型システムでは個々のコンピュータが対等な関係でつながっている。分散される内容も垂直型システムでは階層的に分散化され、水平型システムでは並列的な分散化が行われそれぞれの特徴がでる。

図表VI-10 水平型分散処理システムの例



(3) マルチベンダの分散情報システム事例

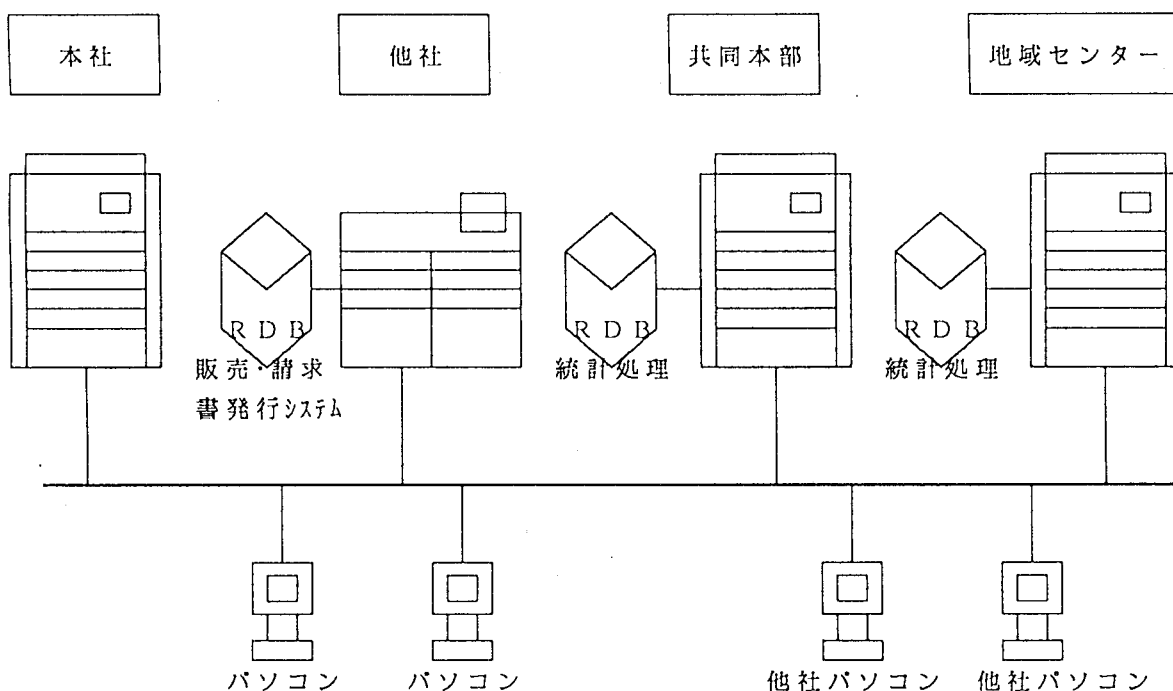
YHPが外資系の食品流通業者のシステムを構成した例を取りあげる。従来のメインフレームによる販売在庫管理システムから、HP9000シリーズを核としたクライアントサーバ型のシステムへ移行した。

販売・在庫・請求業務の効率化、将来のシステムの柔軟な拡張性、流通拠点とのオンラインネットワークの実現可能性を構築の条件とした。4台のサーバを中核にクライアントをパソコンとした。またRDBを採用し、バッチ処理型から分散処理型へ転換を図った。

各端末からオンラインで注文入力すると輸送文書が作られ、自動的にプリントされる。取引先の卸売り業者にもパソコンを導入させ、業者側でも顧客の注文情報を入力させている。また、卸売り業者の顧客へ直接請求書を発行できるシステムになっている。

返品数の激減、在庫管理の合理化で大きなコスト削減を果たしている。応答時間も相当スピードアップした。また、統計データの有効活用も進展した。

図表Ⅵ-11 マルチベンダ分散情報システムの例



3. 分散処理システムにおける処理の構造

(1) 分散処理システムのソフトウェア構造

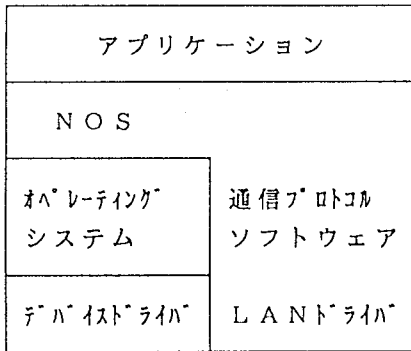
分散処理システムのソフトウェア構造が集中型処理システムのそれと大きく異なるわけではないが、分散管理されているコンピュータ資源をアプリケーションからネットワークを介してアクセスできることが基本である。分散処理システムのオペレーティングシステムは利用者（アプリケーションなど）の必要とする資源が分散管理されているネットワーク上の資源であるか、自コンピュータの直接管理するローカルな資源であるかを判断し、処理の振り分けを行うことが大きな役割である。

分散処理オペレーティングシステムには、ネットワーク機能が一体化されているタイプと、

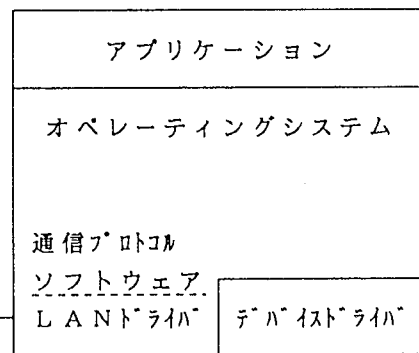
ネットワークオペレーティングシステムと組み合わせて使用するタイプとに分かれる。例えば、UNIXは、TCP/IPを標準の通信プロトコルとして持っており一体型である。現在の商用として普及しているNOSは、最近ではNetWareに絞られているが、かつてはLanManagerもその代表格であった。また、NetWareやLanManagerの持つ通信プロトコルやLANドライバはオペレーティングシステムの中に取り入れられWindows-NT、OS/2に生かされている。

図表VI-12 分散処理オペレーティングシステム

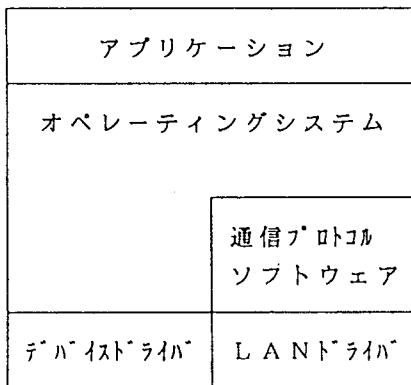
< NetWareに代表されるシステム >



< UNIXに代表されるシステム >



< Windows-NT OS/2に代表されるシステム >



分散処理オペレーティングシステムにおいては、通信プロトコルソフトウェアやLANドライバが非常に重要な役割を果たす。かつてはOSが異なると喋る言葉の種類（即ち通信プロトコルに相当）も違っていた。このため、ネットワーク上でのマルチベンダが実現しにくかった。しかしながら、最近では、オープンシステム化も進んでおり、それぞれのOSが自分が標準に採用するもの以外の通信プロトコルにも対応するようになり、一つのコンピュータに複数個のプロトコルがサポートされる方がむしろ当たり前となってきた。

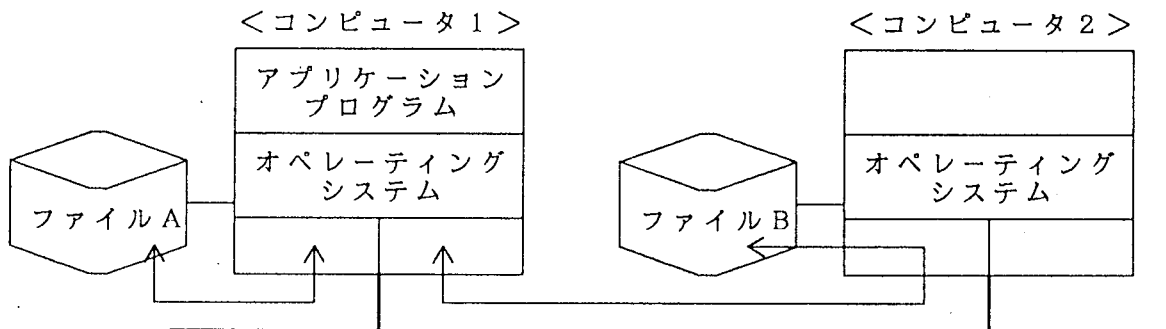
(2) アプリケーションのプラットフォーム

分散システムのネットワーク統合にともないネットワーキングコンピュータという概念が出てきた。アプリケーションプログラムでは自マシンだけでなく、接続されているネットワークの中全体を自分のプラットフォームと想定できる。

① 仮想ファイル

物理的にどのコンピュータに接続されているディスク装置のファイルであるかをアプリケーションプログラムでは意識せずに扱うことができる。

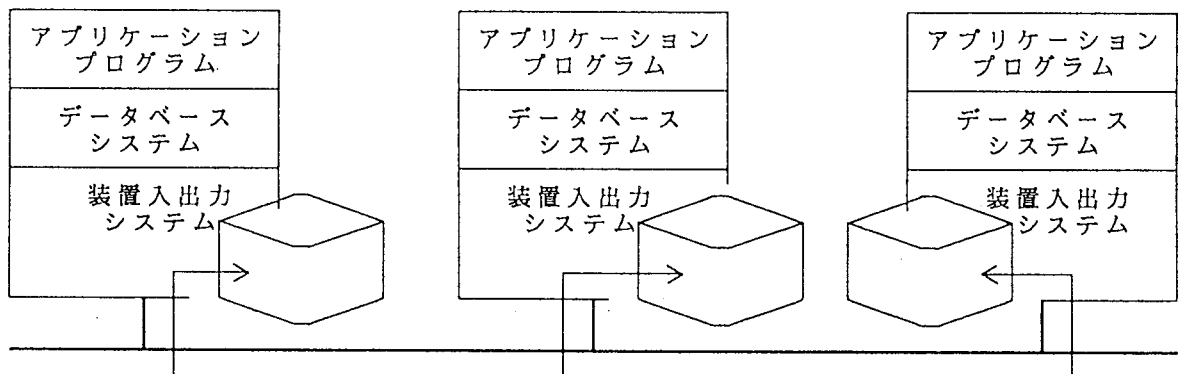
図表VI-13 仮想ファイル



② 分散データベース

物理的にどのコンピュータに接続されたデータベースであるかをアプリケーションプログラムは知らないでアクセスすることができる。データベースはネットワーク内で分散管理されており、どこのデータベースに対するアクセスであっても実際に配置されているデータに対して操作が行われる。

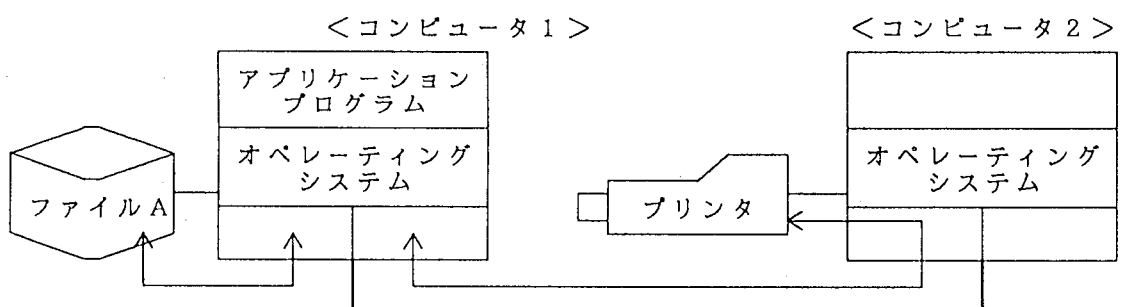
図表VI-14 分散データベース



③ プリンタの共有

物理的にどのコンピュータに接続されているプリンタであるかをアプリケーションプログラムでは意識せずに使うことができる。

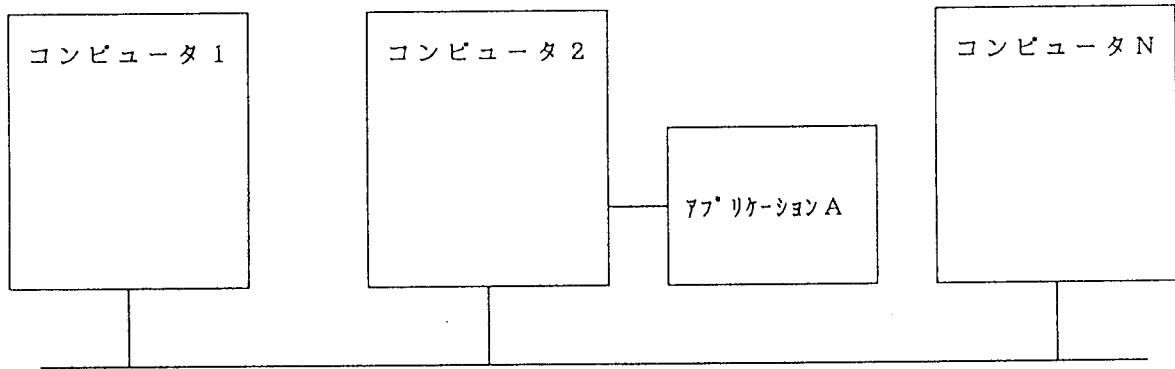
図表VI-15 プリンタの共有



④ アプリケーションプログラムの共有

物理的にどのコンピュータに配置されたアプリケーションプログラムであっても各コンピュータから使用することができる。最近の商用ネットワーク対応ソフトウェアには最大何台までかのコンピュータで同時使用することが認められてきている。このような場合にも使用に際しては、プログラムの使用権を有するかどうかについて確認しておく必要がある。

図表VI-16 応用プログラムの共有



例えばコンピュータ2にインストールされているアプリケーションAを自マシン以外のコンピュータ1やNのユーザが使うことができるが、アプリケーションAを利用者のマシンにロードして動かす場合と、アプリケーションAへの実行条件と依頼事項だけを送ってコンピュータ2で実行を代行してもらう場合とが考えられる。

⑤ 管理情報の共有

ホスト名、ユーザ名、パスワード、ユーザの属性、許される資源の範囲を一元的に管理することとして、管理情報をネットワーク上の各コンピュータに配置することができる。

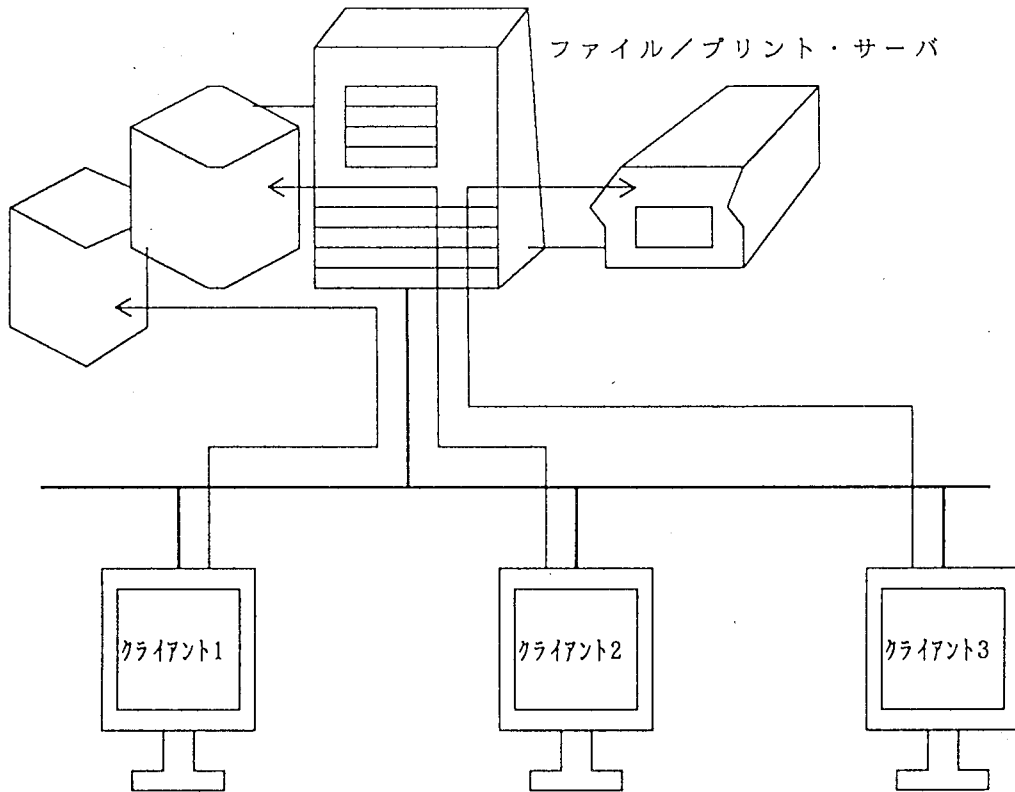
⑥ CPUの共有と負荷分散

パフォーマンスの異なるいくつかのコンピュータがネットワーク上に接続されているとして、利用者が所望のプログラムを、実行要求時点で負荷が軽くかつ適切な実行効率を出せるマシンで動かすことができる(ネットワーク全体の負荷のバランスと実行パフォーマンスを考慮して)。

(3) クライアントサーバ型処理

分散処理システムを構成する代表的な構成手法で、90年代のネットワークコンピューティング環境を構築する最も合理的なものとして進展している。

図表Ⅵ-17 サーバとクライアントの例



上図では、クライアント1とクライアント2のファイルがサーバ上で管理されている。クライアントマシン上で利用者のアプリケーションを実行するときに必要な資源がサーバ上で管理されていると、クライアントマシンはサーバに対して要求を出し、サーバに機能実行の代行をしてもらいその実行結果を受け取ることになる。サーバ側ではクライアントからの要求がいつ到着してもすぐに対応できるよう準備をしている。

(4) ピアツーピア型処理

垂直型の分散処理システムでは階層構造をとり、上位層と下位層とは主従関係を作る場合が多い。これに対して水平型の分散処理では対等な関係で情報交換を行うことが原則で、「ピアツーピア (Peer to Peer) 型の構成」と呼ばれる。これはクライアントサーバ型とは異なる考え方である。ピアツーピア型の分散処理を実現するためにはネットワーク OS を利用することになる。

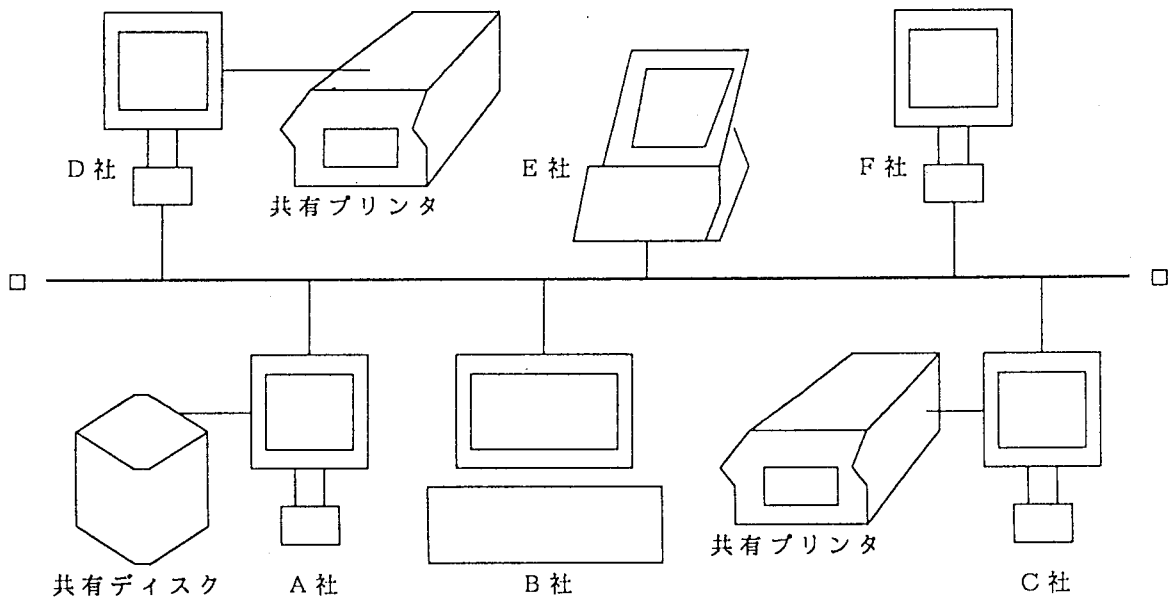
ピアツーピア型の OS は、1 台のマシンでサーバ機能とクライアント機能の両方を同時に内包している。このタイプの処理ではパソコン同士が明示的に 1 対 1 で情報をやりとりする。このネットワークングではどのマシンもサーバになりえ、高価になりがちな専用のサーバ機を用意する必要もなく、システムとしては柔軟性が高いため、クライアントサーバ型に比べて経済的にシステムを組むことが可能である。また、ピアツーピア型のネットワーク OS は比較的規模の小さなシステムに有効である。

このタイプの OS の例として、クライアントサーバ型ネットワーク OS の代表でもある

NetWareのラインアップに追加されたNetWare Liteがあげられる。ネットワーク OSとしての十分な機能性と容易な導入とで小規模な初期のネットワークの制御には最適である。

ネットワーク上にどのようなマシンもサーバ機としてもクライアント機としても接続でき、互いにディスクやプリンタを共有できる。簡単にマルチベンダ環境を構築できるところにも利点がある。

図表VI-18 マルチベンダ型ピアツーピアシステム構成



ピアツーピア型ネットワーク OSには下記のような特徴がある。

- ・ どのパソコンもサーバ機能とクライアント機能の両方を持っており、クライアントサーバ型の OSと異なり、専用サーバ機を必要としない。したがって、サーバ機を高価で規模の大きな構成にしないで済む。
- ・ この型の OSでは、コンピュータ 1 台で他コンピュータからの要求も処理するため、クライアント機能の実行動作のパフォーマンスが落ちることがある。
- ・ インストール、運用が簡単

4. 分散処理オペレーティングシステム

(1) 分散資源管理

分散ネットワークコンピューティング下では利用者やアプリケーションプログラムは資源の物理的な所在位置を意識しないで済むように、分散型のオペレーティングシステムが資源の名前と対応する存在場所との関係をわかる必要がある。

例えば、アプリケーションプログラムがトランスポートアドレスを明示的に指定しなければならないならば、分散システムが再構成されるときにはプログラムも修正しなくてはならない。したがってクライアントのプログラムは使用する資源の所在位置からは独立でなければならない。プログラムは資源の位置を動的に変えることができる。分散型のシステムにおいては、利用者やプログラムで扱う資源名（例えばファイル名）と対応する物理アドレスがネームサーバに記録されている。

ある資源を使いたいクライアントプログラムは、資源名を渡してネームサーバにトランスポート

トアドレスを教えてもらう。したがって、資源を使いたいときは常にそのアドレスを（実際には識別的なコードである。）を用いて要求事項を送る。

(2) ネットワーク機能を持つオペレーティングシステム

分散OSの機能実装はネットワーク機能をベースに作られる。

① UNIXのネットワーク機能

a. UUCP (Unix to Unix Copy)

UUCPは、2地点間でのコミュニケーションを行うための通信パッケージであり、伝送上の誤りチェックや再送機能を持っている。国内外での電話網を用いた広域ネットワークで使われる。

b. TCP/IPは、UNIX標準のプロトコルである。ここで提供されるアプリケーションとしては、

ア. 仮想端末機能を提供するTelnet

端末の物理属性を吸収しリモートマシンとローカルマシンとの間に双方向でのバイト通信機能を提供する仮想端末である。例えばパソコンの端末がホストの端末として動作することができる。

イ. FTP

リモートファイルを自マシンにコピーしたり、リモートファイルのディレクトリを自マシンから操作できる。

ウ. RCP (Remote Procedure Call)

プログラムの種類によって自マシンとリモートマシンとを使い分けられる。RPCでは手続き（関数レベルで）単位でリモートプログラムを呼び出し実行が可能である。が代表的なものである。

② NFS

米国サン・マイクロシステムズ社が開発し普及を図った分散ファイルシステムである。

UNIXにおけるNFSでは他マシンのファイルを自マシンのローカルファイルのように扱える。これは、自マシンのあるディレクトリに他マシンのサブディレクトリをリモートマウントすることにより、利用者はコマンドもしくはプログラムによってそのディレクトリ配下のファイルをローカルアクセスすることが可能となるしなげである。

NFSはUNIXマシン間だけでなく、それ以外のマシン（例えば、MS-DOS, Macintosh, VMSなど）とでも接続が可能である。

③ NIS (Network Information Service)

これもサン・マイクロシステムズ社がNFSと並行して開発したものである。これは一種のディレクトリサービスであり、ネットワーク資源の名前からアドレス情報を提供するシステムで、電話サービスでいう104案内に相当する。

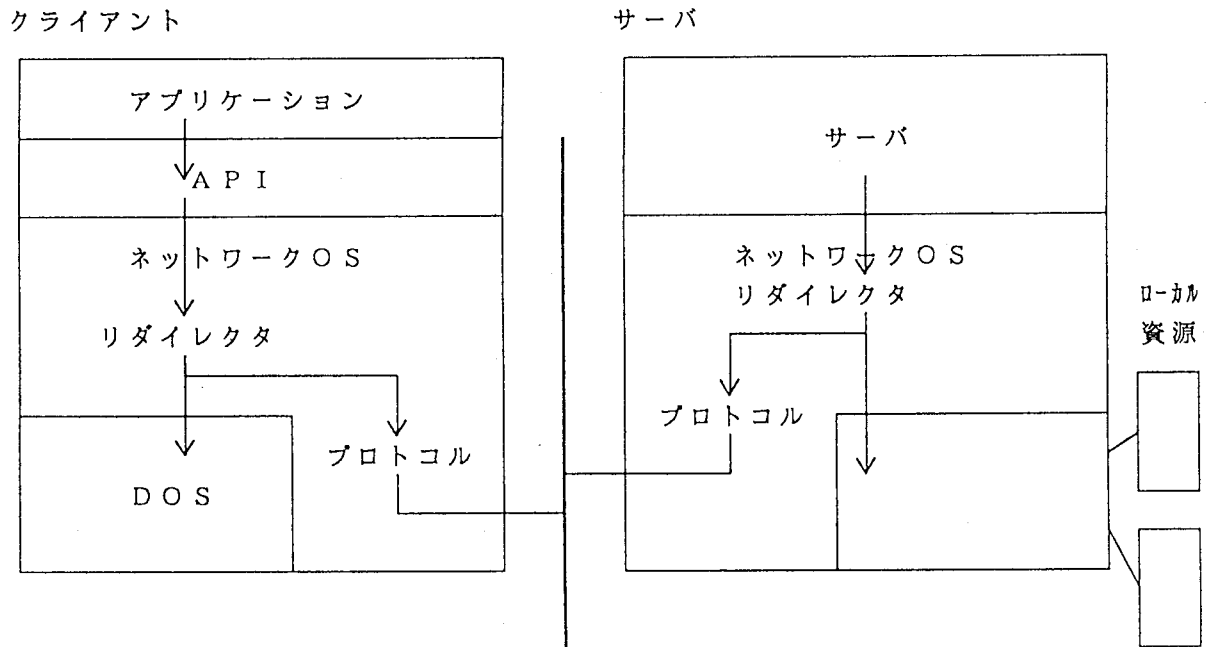
この情報には、パスワード、ホスト名、ネットワークアドレス、ポート番号、通信プロトコルなどが含まれる。NISではこのディレクトリ情報を一元的に管理するが、この情報を複数のマシンで共有するシステム間で一貫性を保たせる工夫を行っている。

NISでは、クライアントがサーバに情報を質問する形態をとる。パフォーマンスを考えると、サーバの情報をコピーしたサブ的なサーバも作れる。

(3) ネットワークオペレーティングシステム (NOS)

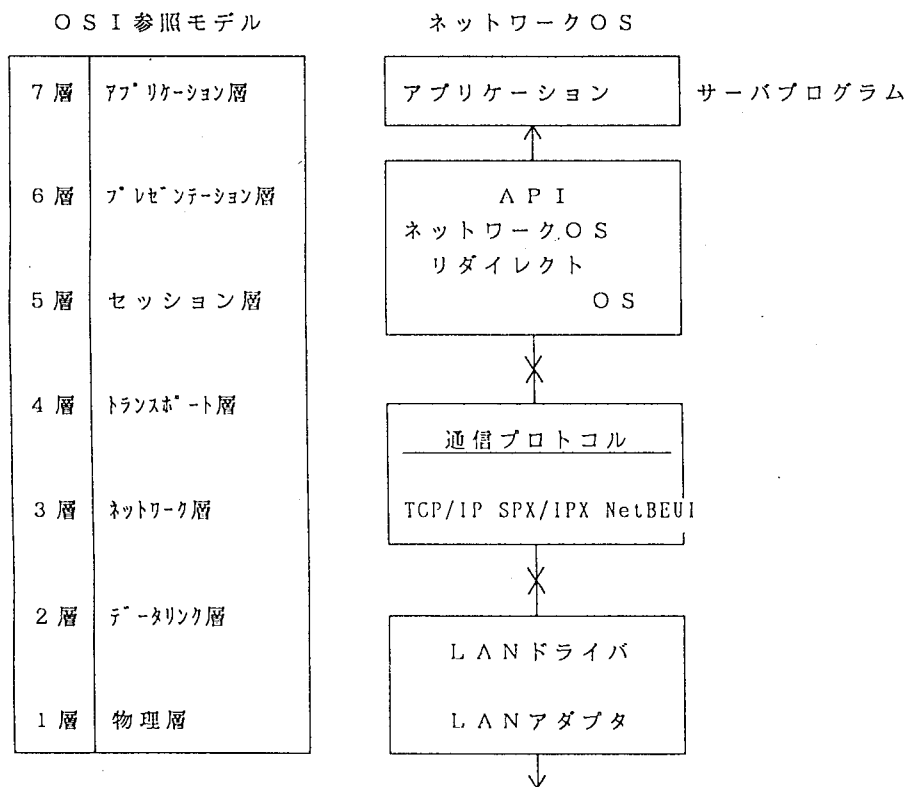
クライアントサーバシステムにおけるNOSを考える。クライアント側とサーバ側との協調動作によってアプリケーションを実現することが基本であり、クライアントマシンにはクライアント用ネットワークOSを、サーバマシンにはサーバ用ネットワークOSを搭載する必要がある。

図表VI-19 ネットワークの基本構成



ネットワークOSのOSI参照モデルとの関係は以下のように捉えられる。

図表VI-20 ネットワークOSとOSI参照モデルの比較



5. 主要用語

NFS	NIS	TCP/IP
アクセス透過性	位置的透過性	アクセス透過性
規模透過性	複製透過性	負荷の分散
処理機能の分散	データの分散	管理の分散
オープンシステム	クライアントサーバ型	ネームサーバ
ネットワークオペレーティングシステム		
ピアツーピア型	垂直分散処理システム	水平分散処理システム
通信プロトコル	分散処理オペレーティングシステム	
分散処理システム	仮想ファイル	分散データベース
Telnet	FTP	RCP
NFS	NIS	NOS
クライアント	サーバ	