高度インテリジェントネットワークにおける
サービス制御ノード構成技術に関する研究

１９９８年１０月

平野　正則

論文内容の要旨

博士論文題名：高度インテリジェントネットワークにおけるサービス制御ノード構成技術に関する研究

学位申請者：平野正則

要旨：

フリーダイヤルや携帯電話等、高機能な電話サービスを効率的に実現することを狙いとした高度インテリジェントネットワークにおいて、カスタマ毎のデータを抱え、呼の接続処理の中心的な役割を果たすサービス制御ノードには、需要に応じた柔軟な構成の実現や、カスタマデータの管理機能の高度化、高い信頼性の達成等が要求される。本論文は、このような要求に的確に対応可能な構成を明らかにするため、多数のモジュールを用いた分散型メモリデータベース構成による、高性能、高信頼で経済的なサービス制御ノード構成技術として、その研究成果をまとめたものであり、以下の８章より構成した。

第１章では、本研究の背景と目的、要求条件と課題、従来研究との関係を述べた。第２章では、本研究で対象とするサービス制御ノードのシステム構成やデータベース配備について述べた。第３章では、サービス制御ノードのデータベースのアクセス特性、モジュールの負荷やその偏り、トランザクションの処理量などを総合的に考慮したデータベースの分散構成方式の評価方法を明らかにし、ファイル共用のデータベース構成に比べ、コスト性能比が２～３倍優れた分散型メモリデータベース構成法を提案した。第４章では、カスタマ毎のサービスデータに対するトラヒックが極めて大きくばらつく場合のトラヒック量の近似方法や、複数モジュールにサービスデータを分散配備した場合のモジュール間の負荷の偏りの評価方法を提案し、サービス制御ノードでは負荷の偏りが１～２割程度に収まることを明らかにした。第５章では、モジュール間の結合機構を機能単位に二重化し、５０モジュール程度を接続した大規模分散構成においてもシステムダウン時間を年間数分以下にできる高信頼構成法や、モジュールの通信処理能力を向上させる周期制御方式における最適な周期時間の決定方法を提案した。第６章では、障害発生からデータベースが回復するまでの時間と、データベースの単位容量当たりのトラヒック量に着目し、メモリデータベースのリカバリ方式の評価技術を明らかにした。これに基づき、サービス制御ノードのリカバリ方式として、半導体ディスク装置にメモリデータベースのログとチェックポイント時点でのデータベースを取得する効率的なリカバリ方式を提案した。第７章では、多数のノードを保守チームが集中して保守する場合について、ノード内、ノード間で相互に冗長化されたシステムの信頼度の時間的な変化に着目し、駆付けの緊急度合いをシステム内の障害装置数や保守チームの到着の有無によりレベル分けすることを提案した。これに基づき、駆付け保守の方法を設定し、信頼性や駆付け回数の評価を通して、急いで駆け付けなければならない緊急駆付け回数の少ない保守方法を明らかにした。第８章では、本研究で得られた結論を総括した。

高度インテリジェントネットワークにおける
サービス制御ノード構成技術に関する研究

目　次

謝　辞

本論文をまとめるに際して、懇切なる御指導ならびに御助言、御尽力を賜った大阪大学大学院工学研究科教授・池田博昌博士に心からの感謝の意を表します。

本論文をまとめる過程に際し、丁重なる御指導、御教示を賜った大阪大学大学院工学研究科教授・前田肇博士に厚く感謝の意を表します。

また、本論文に対して有益なる御討論、御助言を頂いた大阪大学大学院工学研究科教授・森永規彦博士、同教授・小牧省三博士、同教授・児玉裕治博士、同教授・塩澤俊之博士、大阪大学産業科学研究所教授・元田　浩博士ならびに前大阪大学教授・長谷川晃博士（現・高知工科大学教授）に感謝致します。

本研究は、筆者が日本電信電話株式会社（ＮＴＴ）・ネットワークサービスシステム研究所（ＮＳ研）・高機能処理プロジェクト（Ｎ処Ｐ）において、高度インテリジェントネットワークに適用するサービス制御ノードの研究開発の一環として担当したものである。本研究開発の遂行に当たり、御指導、御鞭撻を頂いた、ＮＴＴ・鈴木滋彦取締役（研究開発本部副本部長）、ＮＴＴドコモ・弓場英明取締役、ＮＳ研・研究開発企画部・花澤隆マネージャ、ＮＳ研・Ｎ処Ｐ・田中公紀マネージャ、Ｎ処Ｐ・ＩＮ装置グループ・今川仁グループリーダ、ＮＴＴマルチメディアネットワーク研究所・鈴木孝至部長に厚くお礼申し上げます。本研究の全般に渡って、多くの御討論と御助力を頂いた、東京情報大学・木ノ内康夫教授（当時ＮＳ研）に深く感謝致します。また、本研究を推進するに当たり、御討論頂いた、ＮＴＴアドバンステクノロジ・重松直樹部長（当時ＮＳ研）、ＮＴＴドコモ・上坂久一主幹技師（当時ＮＳ研）、ＮＴＴ研究開発本部・研究開発推進部・寺中勝美部門長（当時情報通信研究所）、ＮＳ研・研究開発企画部・吉見正信主幹研究員、ＮＴＴ光ネットワーク研究所・塩澤恒道主任研究員（当時ＮＳ研）、ＮＴＴグループ企業本部・山根道広担当課長（当時ＮＳ研）、ＮＴＴ情報通信研究所・芳西崇主幹研究員、ＮＴＴ入出力システム研究所・山崎幹夫主任研究員、拓殖大学・小林正光助手（当時ＮＳ研）、ＮＴＴ東京支社・櫻井秀紀主査（当時ＮＳ研）、ＮＴＴ-ＴＥ九州・林誠治氏（当時ＮＳ研）に感謝致します。

最後に、これまで様々な面から御指導、御鞭撻を頂いた、大阪大学・橋本昭洋教授、国際電気通信基礎技術研究所・酒井保良副社長、東京工科大学・松永俊雄教授、ＮＴＴアドバンステクノロジ・平松琢弥部長、ＮＴＴエレクトロニクス・多嶋清次郎副事業部長、ＮＴＴ情報通信研究所・武井安彦部長に心からお礼申し上げます。

第１章　　序論

１.１　背景と目的

高度情報化社会に向け通信網の高度化、多様化が急速に進展してきている。電話サービスにおいても、電話機の設置場所に係わらず設定された論理的な番号に基づいて接続するサービスや、携帯電話のように電話機の位置を絶えず追跡して位置情報をネットワーク内に登録しておき、電話機の位置に係わらず通話ができるサービスなどが普及してきている。このようにネットワークでカスタマ毎のデータを保持し、この内容に基づいて多様なサービスを実現する網がインテリジェントネットワーク（IN：Intelligent Network）である。我が国では、１９８５年にサービスが開始されたフリーダイヤルを始めとして、携帯電話、ダイヤルＱ^{2}、仮想私設網（VPN : Virtual Private Network）、ＰＨＳ（Personal Handy-phone System）等のサービスが提供され、インテリジェントネットワークを用いた電話サービスの多様化が進んでいる。

高度インテリジェントネットワーク（高度ＩＮ）は、このような状況に対応するため、多様なサービスをタイムリかつ効率的に提供することや、カスタマの要望に応じたサービス内容の変更を容易とすること等を狙いとした次世代のＩＮであり、世界各国で研究、開発が進められている[(1)~(4)]。この高度ＩＮでカスタマデータを抱え、呼の接続処理の中心的な役割を果たすノードがサービス制御ノード（SCP：Service Control Point）であり、需要に応じた柔軟な構成の実現や、カスタマデータの管理機能の高度化、高い信頼性の達成等が要求される[(5)]。

本論文は、このような要求に的確に対応可能な構成を明らかにするため、需要の変動や機能要求へ柔軟に対応する分散処理化、サービス毎カスタマ毎のデータを効率的に管理するデータベース化、運転保守の効率化とのバランスを考慮した経済的な高信頼化の各面からアプローチしたものであり、多数のモジュールを用いた分散型メモリデータベース構成による高性能、高信頼で経済的なサービス制御ノード構成技術として、その研究成果をまとめたものである。

1.2　研究の位置付け

電話サービスを行うための電話網は、家庭や事務所等に置かれる電話機を接続収容する交換機、および交換機間を相互に接続する伝送路等から構成される。日本全国にはおおよそ四千台の交換機が設置されている。電話機に付与される電話番号は、それを収容する交換機の位置や、交換機内での収容端子等の物理的な位置により一義的に決められている。

電話の接続は、発信者が着信者の電話番号をダイヤルすることにより行われる。まず、発信者がダイヤルを行うと、発信者の電話機を収容している交換機がダイヤルされた番号を受け取る。この番号から着信者の電話機を収容している交換機が一義的に決まるため、発信側の交換機から着信側の交換機までの伝送路を確保し、ダイヤル番号を着信側の交換機に渡す。着信側の交換機では、ダイヤルされた番号に対応する端子に接続された電話機に呼び出しをかけ、着信者が受話器を取ると、通話が出来るようになる。このように、従来の電話サービスの場合は、電話番号により、どの交換機のどの端子位置に接続された電話機に対する呼であるかが一義的に決まる。

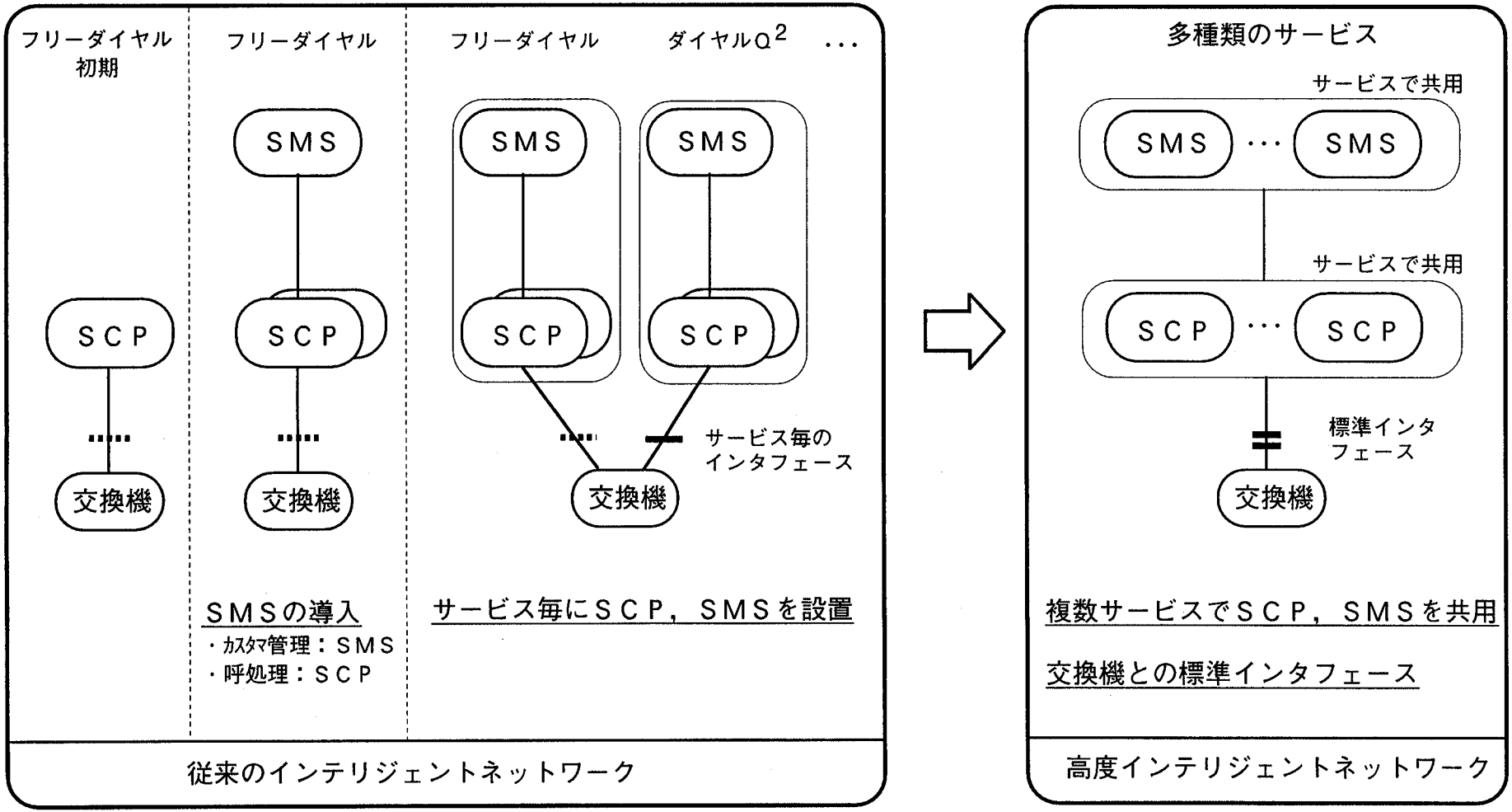
一方、フリーダイヤルサービスの場合は、電話機を収容する交換機の設置位置や端子位置によって決まる物理的な番号ではなく、論理的な番号に基づいて接続する必要がある。このため、ダイヤルされた論理番号を物理番号に変換するデータを蓄え、検索するためのデータベース機能が必要とされる。これらの機能を四千台の交換機に持たせることは維持管理が大変であるため、交換機とは別のサービス制御ノードで実現する方法がとられている。交換機や伝送路からなる従来の電話網に、データベース機能を実現したサービス制御ノードを付加した網をインテリジェントネットワークと言う。インテリジェントネットワークでは、交換機や伝送路を伝達レイヤ、サービス制御ノードを高機能レイヤと言う。

インテリジェントネットワークを用いた最初の電話サービスは、１９８５年にサービスが開始されたフリーダイヤルである[(6)][(7)]。フリーダイヤルの初期には、高機能レイヤはサービス制御ノード（ＳＣＰ）のみから構成され、サービス処理に必要な情報等は直接、端末からＳＣＰへ投入する方法で行われていた。また、顧客情報や空

き番号等は、支店や営業所の独自のデータベースで管理されていた。このため、サービス申し込みを受け付けた時に割り当てた空き番号等、サービス処理に必要な情報を、サービス開始に合わせてＳＣＰへ投入する必要があり、カスタマの申し込みからサービス開始までの業務が煩雑であった[8]~[10]。その後、カスタマ管理の効率化のため、１９８９年にサービス管理ノード（ＳＭＳ）が導入され、カスタマ情報のデータベース化が行われた。これにより、ＳＭＳに接続された端末から空き番号の割当てや、サービス開始に合わせて、呼処理に必要な情報のＳＣＰへのダウンロードが自動的に行われるようになった[11]。また、ＳＭＳの導入によりカスタマ自身によるサービス条件の変更も可能となった。この時点で、カスタマ情報やサービス条件等をサービス管理ノードで一元的に管理し、呼処理に必要な番号変換情報等がＳＭＳからＳＣＰへダウンロードされるようになり、カスタマ管理はＳＭＳ、呼処理はＳＣＰで機能分担する形態が確立された。その後、携帯電話やダイヤルＱ^{2}等、サービスの多様化に伴い、サービス毎にＳＭＳとＳＣＰが構築されていった[12]。この段階では、ＳＣＰはデータベースアクセス機能のみを有し、サービス制御機能は交換機側にあり、サービスの新規導入には交換機側のソフトウェアの変更が生じ多大の工数と期間が必要であった[8][13][14]。

高度インテリジェントネットワーク（高度ＩＮ）は、今後のより一層のサービスの多様化や高度化に柔軟に対応するため、需要の異なる多種多様なサービスを効率的に提供することや、新しいサービスを短期間に提供することを狙いとした次世代のインテリジェントネットワークである。高度ＩＮでは、複数のサービスでＳＣＰやＳＭＳを共用できることや、交換機とＳＣＰはサービスに依存しない標準インタフェースで接続し、新サービスに伴う交換機側での機能追加を不要とすることが要求される。また、ハードウェア構成上は多様なサービスに柔軟に対応するため、サービスに依存しないＳＣＰ構成や、プロセッサの追加により処理能力が容易に拡張できること、さらに高い信頼性の達成等が望まれる[5][8][14]。

インテリジェントネットワークの初期から高度インテリジェントネットワークに向けた進展の状況を図１.１に示す。



SCP：サービス制御ノード，　　SMS：サービス管理ノード

図1.1　　インテリジェントネットワークの進展状況

高機能な電話呼の接続処理がインテリジェントネットワークでどのように行われるか、図１.１を用いて、以下で簡単に説明する。まず、電話機から投入された電話番号は交換機で解読される。その結果、電話番号が０１２０で始まるものであればフリーダイヤルであることが判明し、フリーダイヤルに加入しているカスタマのサービスデータを収容しているサービス制御ノード（ＳＣＰ）に対して０１２０の後に続く６桁の論理番号を送る。１台のＳＣＰにカスタマデータを収容できない場合は、複数のＳＣＰに分散収容されることとなり、論理番号を送るとき、当該カスタマのサービスデータがどのＳＣＰに収容されているかの判断が必要となる。一般に、カスタマデータの収容に当たっては、６桁の論理番号の上位１～２桁によりどのＳＣＰに収容するかが決められており、上位１～２桁の解読によりどのＳＣＰに収容されているかが判明する。ＳＣＰでは論理番号と物理番号の対応表をデータベースとして格納しており、このデータベースにアクセスし、論理番号を物理番号に変換して交換機に送る。交換機では送られてきた物理番号に基づいて電話呼の接続処理を行う。このとき、フリーダイヤルサービスであることから、通常の電話呼とは異なり課金は着信側に行う。交換機で解読した電話番号が０９９０で始まるものであればダイヤルＱ^{2}であると判断し、ダイヤルＱ^{2}に加入しているカスタマのサービスデータを収容しているＳＣＰに問い合わせをかけ、同様に物理番号を得て、当該物理番号に基づいて接続処理を行う。このように、サービス毎にどのＳＣＰに問い合わせを行うか等の制御や、サービスに基づく課金処理等は交換機側で行っており、新しいサービスの追加やサービス内容の変更およびＳＣＰの増設等に伴い、交換機側のプログラムの変更が必要となる。

一方、高度インテリジェントネットワークでは、入力された電話番号を解読するところまでは同様に行うが、電話番号が０１２０や０９９０で始まるなど、高機能な電話サービスであることを検出すると、具体的なサービス内容が何であるかを判断することなく、あらかじめ決められたＳＣＰに問い合わせをかける。ＳＣＰ側では、サービス種別を判断すると共に、当該サービスを受けているカスタマのサービスデータがどのＳＣＰに収容されているかを判断し、当該ＳＣＰにこの問い合わせを転送する。以下、呼の制御はカスタマのサービスデータを収容しているＳＣＰで行われる。ＳＣ

Ｐから交換機への指示も、たとえば、物理番号に接続し、課金を着信側に行う等、サービスに依存しない指示で行われる。このため、新しいサービスの追加やサービス内容の変更等が生じたとしても交換機側のプログラムの修正が不要となる。これにより新しいサービスを短期間で提供することや、サービス内容をカスタマ毎に設定することが容易となる。

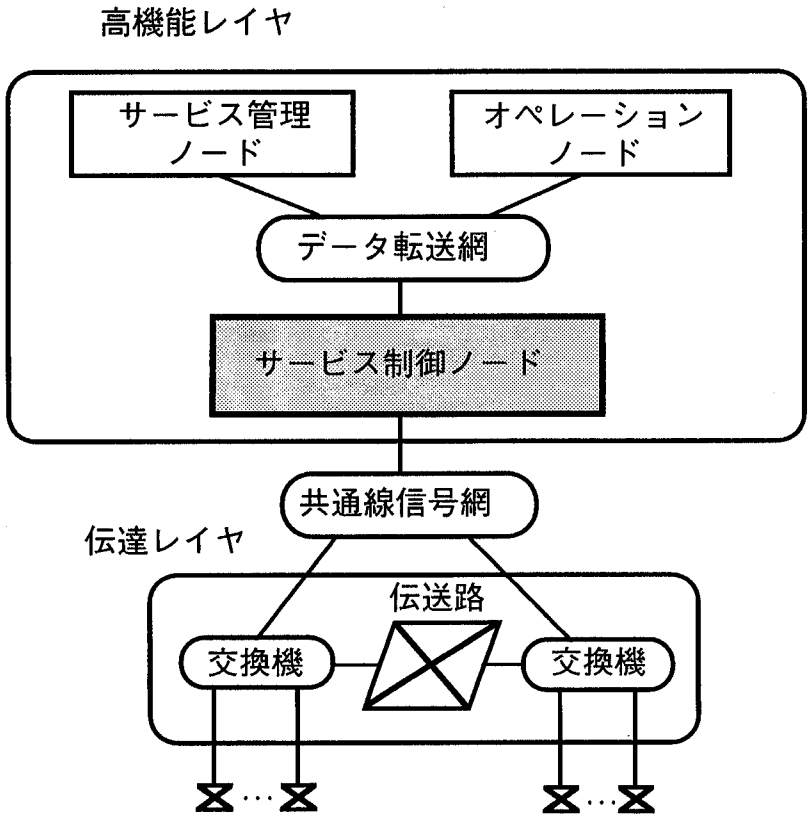
高度インテリジェントネットワーク（高度ＩＮ）においても、従来のインテリジェントネットワークと同様に、伝達レイヤと高機能レイヤの二階層で構成され、サービス制御ノードと交換機は共通線信号網で接続される。高機能レイヤはサービス制御ノード、サービス管理ノード、およびオペレーションノードから構成される[15]。サービス制御ノードは交換機と接続され、電話呼の接続処理に直接係わるノードであり、リアルタイム性の高いデータベースアクセス機能が要求される。サービス管理ノード（ＳＭＳ）はカスタマの情報や、そのカスタマが受けるサービス内容等をデータベースとして管理し、サービスオーダの受け付けや、サービス内容の変更等を行う。また、ＳＭＳで管理する情報のうち呼処理に必要な情報のみをＳＣＰへダウンロードする[16]。オペレーションノードは運転保守を行うノードであり、多数のＳＣＰの障害処理や、ソフトウェアの変更に伴うファイル更新処理等を行う[17]。

サービス制御ノードの実現技術としては、ハードウェアの構成技術、ソフトウェアによる制御技術があるが、本論文では、サービス制御ノードのハードウェア構成技術を対象とする。本研究の位置付けを図１.２に示す。

１.３　要求条件と課題

１.３.１　サービス制御ノード実現に向けた要求条件

サービスの多様化やサービスのカスタマイズ化に柔軟に対応するため、サービス制御ノードに要求される条件は以下の通りである。



高度インテリジェントネットワーク

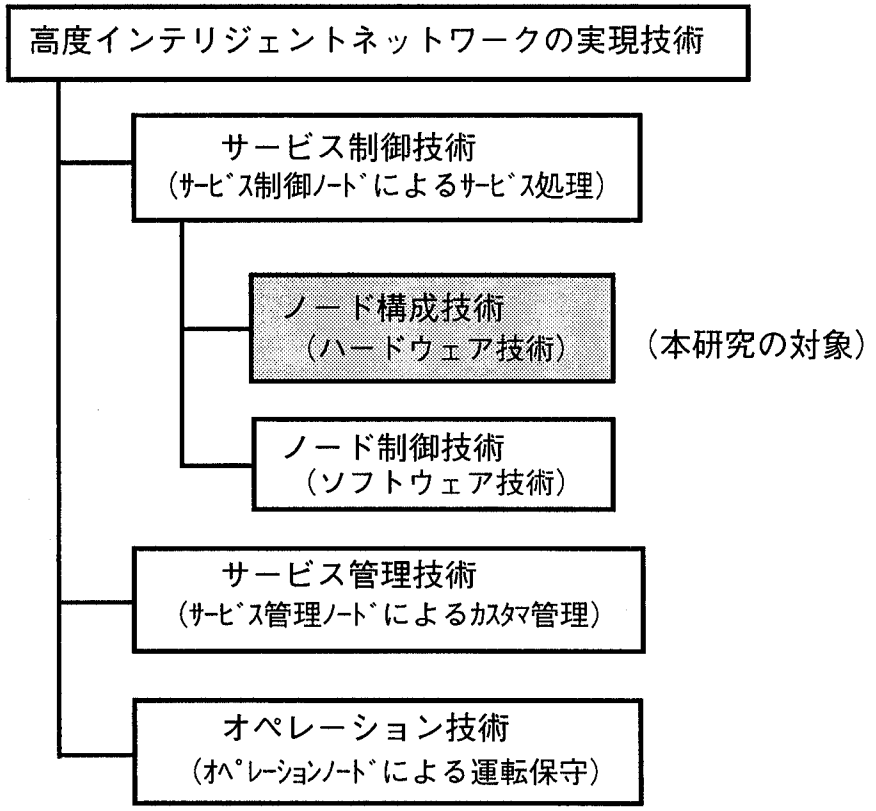


図１.２　研究の位置付け

（１）需要に応じた柔軟な構成の実現

高度ＩＮでは、多様な内容のサービスが展開されるだけでなく、ノード規模は小規模から大規模まで幅広く分布する。またサービス毎の規模も事前に予測することが難しく、状況に応じ、即座に処理能力が拡大されなければならない。このため、分散処理構成による拡張性の高いシステムの実現が重要となる。

（２）データ管理機能の高度化

呼処理の過程で多様なデータを管理する高度ＩＮでは、今後、サービスの高度化、サービス種類の増加等によりデータの管理は一層複雑となる。これに対応するためにはデータベース技術の適用が必要となる。銀行のオンラインシステムやクレジット照会システム等のオンライントランザクション処理（OLTP：On-Line Transaction Processing）システムの分野ではデータベース技術は既に用いられているが、応答時間条件の厳しい高度ＩＮに導入するに当たっては、高速化のためデータベースの全てを主メモリや半導体ディスク装置等、半導体メモリ素子を用いた記憶媒体に格納することが必要となる。ＯＬＴＰシステムにおいても半導体メモリ素子の高集積化の進展と相まって、データベースの全てを主メモリ等に格納する方法の研究は活発に行われているが、分散処理時の望ましい構成は明らかにされていない。

（３）信頼されるサービスの実現

高度ＩＮサービスは伝達レイヤと高機能レイヤにより実現され、両レイヤが同時に機能するため、通常の電話サービスに比べて、一般的にアベイラビリティは低下する。しかし、その低下度合いは、ユーザからみて気づかない程度とすることが必須である。また、大規模なシステムになるほど、システムダウン時の影響が広く大きくなるため、一層の高い信頼性が要求される。一般的に、分散構成下では冗長化によりアベイラビリティを向上させることは容易であるが、同時にコストアップを招きやすい。このため、経済的に高信頼化をはかることが重要となる。

さらに、高度ＩＮでは、時々刻々変化するカスタマのデータをもとにサービスが行

われるため、データの喪失、破壊を防ぐこと、すなわち、ハードウェア障害、ソフトウェアのバグに対するデータの回復を保証することが併せて重要となる。

本論文では以上の要求に対応するため、分散処理化、データベース化、高信頼化の各面からアプローチし、拡張性に優れ、信頼性の高いサービス制御ノードを経済的に実現するハードウェアシステム構成技術の確立を目的としている。本研究へのアプローチ方法を図１.３に示す。

１.３.２　サービス制御ノード構成の課題

望ましいサービス制御ノードを実現するための主要な課題は以下の通りである。

（１）データベースの分散構成技術

効率の良い分散型のサービス制御ノードを実現する上では、どのようにデータベースを配置、分散するかが重要な課題となる。なかでもデータベースを格納する記憶媒体や、複数のモジュールからのデータベースの共用形態等は、応答時間、処理の効率、さらにはデータの信頼性に影響するため、これらの選択は大きな課題となる。また、カスタマデータを分散する場合、それぞれのデータへのトラヒックのばらつきに起因してモジュール間の負荷の偏りが発生する。分散処理システムの方式設計や設備設計を効率的に行う上で、この偏りの評価や、偏りを平準化しモジュールの稼働率を高めることが課題となる。さらに、複数のモジュールを相互に結合した分散処理システムでは、結合機構がシステム全体の信頼性や性能を決定することになる。このため、結合機構の高信頼化、高性能化が重要な課題となる。

（２）データベースの高信頼化技術

データベースを効率よく高速に処理するためには、データベースの全てを主メモリ上に常駐するメモリデータベースの適用が有効である。しかし、メモリデータベース

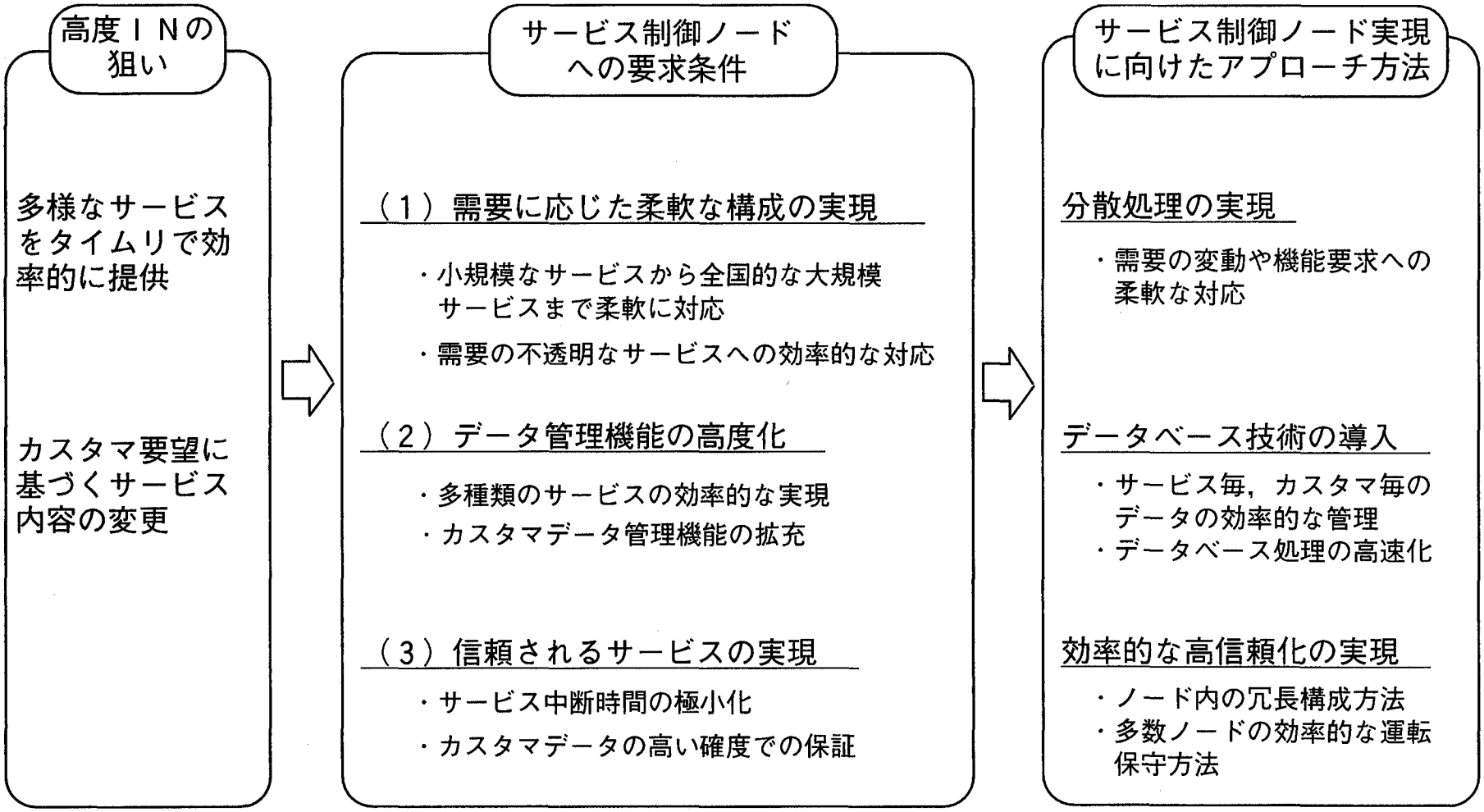


図１.３　研究へのアプローチ方法

はソフトウェアのバグ、ハードウェア障害及び電源障害により、主メモリ上のデータベースが破壊、喪失されやすいという欠点を併せ持っている。このため、あらかじめデータベースの回復に必要な情報を障害に強い記憶装置上に格納しておき、障害時にはこの記憶装置上の情報をもとに障害直前のデータベースを効率よく復元する技術が要求される。

（3）高信頼化運転保守技術

全国的な高度ＩＮサービスを展開するためには、ネットワーク内に多数のサービス制御ノードを広域的に配置する必要がある。また、地震、水害等の大規模災害に対応するためには地理的に離れた地点に設置されたノード間での相互バックアップが必須となる。このように相互にバックアップされたサービス制御ノードに要求される信頼性を確保した上で、多数のノードの運転保守を可能な限り効率化、経済化する方法が要求される。

本研究における課題の概要を図１.４に示す。

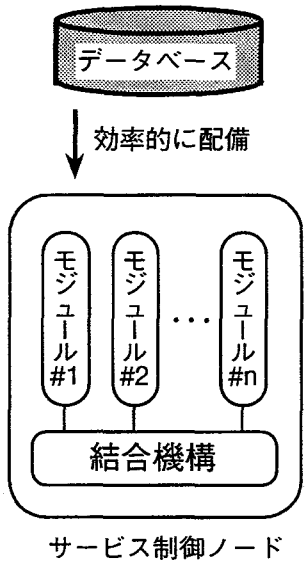
１.４　従来研究と本論文の関係

本節では、本論文で扱う課題に対して、従来の研究との関係を示す。

１.４.１　データベースの分散構成技術との関係

多数のモジュールによるデータベースの分散構成を効率的に実現するためには、高性能で高信頼なデータベースの分散処理方法や、モジュール間の負荷の偏りの評価方法、高信頼で拡張性の優れたモジュール間の結合方法を明らかにすることが重要となる。それぞれについて、従来技術との関係について述べる。

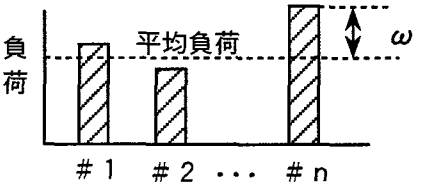
（1）データベースの分散構成技術



・サービス制御ノードに適したデータベース（ＤＢ）の分散処理方法

（・ＤＢを格納する記憶媒体（主メモリ／磁気ディスク装置）
・複数のモジュールからのＤＢの共用形態）

・モジュール間の負荷の偏り評価

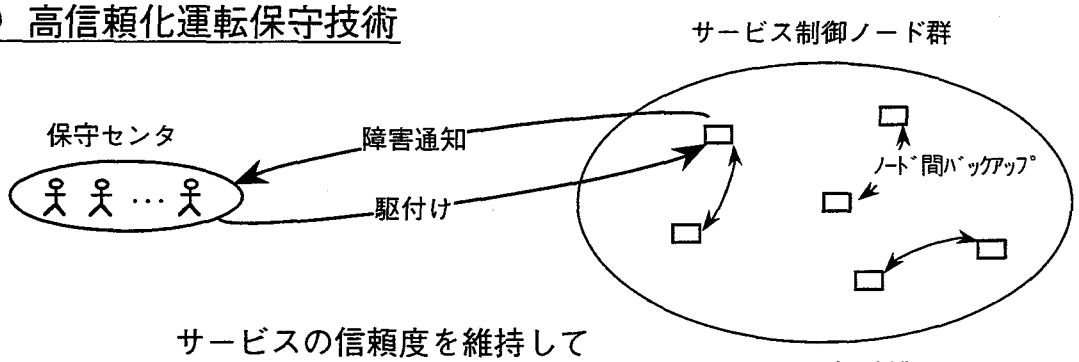


・高信頼、高性能なモジュール間結合方法

（2）データベースの高信頼化技術

・ソフトウェアのバグ，ハードウェア障害によるＤＢ破壊
・電源障害によるＤＢ喪失

（3）高信頼化運転保守技術



サービスの信頼度を維持して運転保守コストを軽減

□：サービス制御ノード

図１.４　課題の概要

（1）データベースの分散処理技術との関係

高度ＩＮ用ＳＣＰの実現法について、トラヒックの増大に対しては負荷分散により対応する構成の提案[18][19]はあるが、機能分散と負荷分散を統合した分散構成の提案や、分散処理を採用する上での課題とその対応策については明らかにされていない。分散構成の実現に当たっては、高度ＩＮに適したデータベースの分散処理の方法、高信頼化構成法を明らかにすることが主要な課題となる。ＯＬＴＰ（On-Line Transaction Processing）システムの分野ではデータベースを磁気ディスク装置に配備し、複数のモジュールで分散処理する方法はすでに実現されている[20]～[22]。しかし、トランザクション処理の高速化のため、データベースのすべてを主メモリに常駐したメモリデータベースについての分散処理の実現法に関する報告例はない。また、高信頼化に関しては、地震、火災等の災害に対してもサービス中断を生じないように、地理的に離れた地点でのバックアップ方法の提案[23][24]はあるが、ノード内を分散構成とし、さらにノード間で相互バックアップしたシステムの効率的な冗長化構成法についての提案はない。

本論文では、小容量でアクセス頻度（単位時間当たりのアクセス回数）が極めて高いこと等、サービス制御ノードのデータベースの特性を明らかにし、それに基づいて、複数のモジュールからのデータベースの共用の可否、主記憶／半導体ディスク装置いずれかの格納媒体の選択の２点に注目して構成案を設定した。これら案について、モジュール間の負荷の偏り、データベース処理でのダイナミックステップ数に対する評価を加え、コスト、性能面から総合的に比較を行った。結果をサービス制御ノードに適した分散型メモリデータベース構成として提案する。また、モジュール間バックアップとノード間バックアップを組み合わせた効率的な冗長構成法を提案する。

（2）モジュール間の負荷の偏り評価技術との関係

高度ＩＮサービスでは、カスタマ毎のトラヒックが大きくばらつく。トラヒックの高いカスタマとトラヒックの低いカスタマとでは、数千倍から数万倍のトラヒックの差がある。また、あるカスタマへのトランザクションは、当該カスタマのサービスデ

ータを用いて処理され、トランザクションとサービスデータが結びついている。このようにトラヒックのばらつきの大きいサービスデータを多数のモジュールに分散配備して、トランザクションを分散処理する場合、トランザクション処理の実行はサービスデータにアクセス可能なモジュールに限定されるため、モジュール間の負荷の偏りが問題となる。

一般に、分散処理システムの負荷の偏りに関する研究は、モジュールに配置されるカスタマ毎のサービスデータとトランザクション処理の結びつきの有無によって大きく2つに分類できる。サービスデータとトランザクション処理が結びつかない例としてパケット交換などの通信処理や音声、画像などのメディア変換処理があり、1つ1つのトランザクションの処理量（ダイナミックステップ数）のばらつきは大きいが、どのモジュールでも処理が可能である。このようなシステムでは、モジュールにいかに効率よくトランザクションを割振るかが大きな課題であり、これに関しては、国内外で種々研究がなされている[25][26]。一方、サービスデータとトランザクション処理が結びつく例として、高度ＩＮシステムや、銀行のオンラインシステム、クレジット照会システムのような多くのＯＬＴＰシステムがあり、カスタマ毎のサービスデータへのトラヒックのばらつきが大きく、トランザクション処理の実行はサービスデータにアクセス可能なモジュールに限定される。従来から、サービスデータを磁気ディスク装置などの外部記憶装置上に集中して格納し、各モジュールで共用する方法がある[27]。この場合は、どのモジュールも、全てのサービスデータに対するトランザクション処理が可能であり、モジュール間の負荷の偏りは生じない。しかし、近年、メモリ素子の高集積化を背景にトランザクション処理の高速化やスループット向上のため、サービスデータからなるデータベースの全てを主メモリ上に常駐するメモリデータベースが採られるようになってきた[28][29]。メモリデータベースではサービスデータの集中による共用は困難である。多数のモジュールでデータベースを非共用とした場合にはモジュール間の負荷の偏りが生ずるが、これに関する研究はほとんどなされていない。

本論文では、以上のようにトラヒックのばらつきの大きいサービスデータを多数の

モジュールに分散配備し、トランザクションを分散処理するシステムを対象として、カスタマ毎のサービスデータに対するトラヒック比が数千から数万倍と大きくばらつく場合のトラヒック量の近似方法を提案する。これに基づき、複数のモジュールにサービスデータを分散配備した場合の、モジュール間の負荷の偏りの評価方法、上限値の推定方法を明らかにする。

（3）モジュール間の結合技術との関係

高度ＩＮのＳＣＰでは、小規模なサービスから全国的な大規模サービスまで柔軟に対応することや需要の不透明なサービスへ効率的に対応することが要求される。このような需要の変動にタイムリに対応可能とするためには、多数のモジュールからなる大規模な分散構成によりＳＣＰを実現することが有効である[5]。分散処理システムを拡張性良く高信頼に実現するためには、モジュール間を接続する結合機構のデータ転送能力に制約が生ぜず高い信頼性が達成されることが要求される。また、結合機構の高信頼化に加えて、モジュール個々には高い通信処理能力が要求される。トランザクション処理の場合は、モジュール間で送受されるデータ長は数１００バイトと短いが、送受を高頻度で行う必要がある。これを実現するため、通信処理を専用に行うハードウェア（以下、通信制御チャネルと記す）を適用し、この通信制御チャネルの単位時間当たり処理可能な通信回数を向上させることが重要となる。従来から分散処理システムのモジュール間の結合方法として、独自のＬＡＮを用いたものや[22]、業界標準のイーサネット、ＦＤＤＩ（Fiber Distributed Data Interface）を用いたものがＯＬＴＰシステム等で提案されている。これらのシステムで用いられているＬＡＮの最大データ転送速度は１０～１００ Mbps 程度であり、今後のプロセッサ性能の向上や接続するモジュール数の増加を考えると必ずしも十分とは言えない。モジュールの追加により柔軟に処理能力を拡張可能とするためには、高いデータ転送能力を有するスイッチを適用する方法が有効となる。しかし、スイッチを適用する場合は、その信頼性がノード全体の信頼性を決定することとなり、スイッチの高信頼化が必須となる。パソコン等を相互に接続した企業内のＬＡＮを構築する場合のバックボーンとして、

既にスイッチが適用されている[30]が、高い信頼性が要求されるオンライントランザクション処理システムへの適用例は少ない。

本論文では、分散構成を採るＳＣＰ内のモジュール間を結合する方式として、十分なデータ転送能力を確保することからスイッチ方式とし、モジュールとのインタフェースはＡＴＭを適用する分散処理システムを対象として、結合機構の高信頼化構成法や、結合機構に接続されるモジュールの通信処理能力を向上させる制御方式ついて提案する。

1.4.2　データベースの高信頼化技術との関係

サービス制御ノードのデータベースは容量が比較的小さく、アクセス頻度が高い特徴を有し、トランザクション処理性能の向上のためにはデータベースの全てを主メモリ上に常駐するメモリデータベースの適用が有効である。しかし、メモリデータベースは、ソフトウェアのバグ、ハードウェア障害および電源障害により、主メモリ上のデータベースが破壊、喪失されやすいという欠点を併せ持っている。このため、あらかじめデータベースの復元に必要な情報を障害に強い記憶装置上に格納しておき、障害時にはこの記憶装置上の情報をもとに障害直前のデータベースを復元する方式を併せて採用する必要がある。この実現のため、Ｌｅｈｍａｎ、Ｅｉｃｈらは磁気ディスク装置上にデータベースの更新ログ（ＬＯＧ）とチェックポイント時点のデータベース（ＣＰＤＢ）を取得し、障害時にはＬＯＧとＣＰＤＢから障害直前のデータベースを復元する方式を提案している[31]~[33]。磁気ディスク装置の入出力ボトルネックを解消するため、不揮発化したバッファメモリにＬＯＧをある程度バッファリングし、一括して磁気ディスク装置に書込む方式を採っている。しかし、この方式では電源障害時にバッファリングされているＬＯＧを保持することは可能でも、ソフトウェアのバグによりバッファメモリ上のＬＯＧが破壊される可能性がある。一方、高倉らは磁気ディスク装置を用いず、読出しのデータ転送速度がより高速で不揮発性を持つフラッシュメモリ上にＬＯＧとＣＰＤＢを取得し、データベースの復元を高速に行う

方式を提案している[34]。主メモリを二重化するとともに、一時的に更新内容を保持するバッファを設けるなど主メモリを専用化した構造としている。このため、汎用的な装置の適用による経済性の追求やソフト制御の簡潔化が期待しにくくなるという問題がある。

本論文では、これらの要因を回避するため、汎用性の高い装置構成を前提に、ＬＯＧを主メモリでバッファリングせずトランザクション毎に外部の記憶装置に書込むこととしたメモリデータベースの復元方式（以下、リカバリ方式と記す）を検討する。この場合、外部の記憶装置の形態はＬＯＧを単に記憶するパッシブな方式とＬＯＧをプロセッサで処理可能なアクティブな方式に２大別される。前者を代表し高速な入出力が可能な半導体ディスク装置を適用する方式を設定し、後者については別のモジュールのプロセッサに付属する主メモリを適用する方式を代替案として設定する。２つの代替案について、障害発生からデータが回復するまでの時間と、データベースの単位容量当たりのトラヒック量に着目し、メモリデータベースのリカバリ方式の評価方法を明らかにする。これに基づき、サービス制御ノードのメモリデータベースのリカバリ方式として半導体ディスク装置を用いた方式を提案する。また、劣化判定機能を有する保守の容易なバッテリを用いた、半導体ディスク装置の効率的な不揮発化方法ついて提案する。

１.４.３　高信頼化運転保守技術との関係

高い信頼性が要求される交換機等では、従来からノード内で二重化する冗長構成が一般にとられている。しかし、データベースを用いて全国を対象にサービスを行う高度ＩＮのサービス制御ノードでは、ノード内の二重化のみでは、地震、火災等の大規模災害に対しては十分でなく、地理的に離れた２つの地点にノードを設置して、ノード相互でデータベース上の情報を含めてバックアップする冗長構成がとられるようになってきている[23]。同様に、バンキングシステムやクレジット照会システム等のオンライントランザクション処理システムにおいても、サービス内容の多様化、高

度化に伴って、ノード内で二重化し、さらに地理的に離れて設置された２つのノード間で相互バックアップする方式が採用されてきている[24][35]。

一方、システムの信頼性は、このような高信頼化のための冗長方式とならんで保守方式により大きく変動する。ノードに保守チームが常駐している場合は、障害検出と同時に修理に取り掛かることが可能である。しかし、ノード毎に保守チームを常駐させると保守コストの増大を招くため、通常、保守センタに保守チームを集中し、複数のノードを少数のチームで保守する方法が採られている[36]。この場合、サービス停止につながるような障害では、常駐の保守チームにより緊急に駆付けて対応し、サービス停止に至らない軽度の障害では、ある程度時間的な余裕を持って平日勤務の保守チームにより対応する等、システムの緊急度合いに応じて駆付け時間を調整することが多い。渡辺は、二重化した交換機と宅内装置からなる加入者系を対象として、駆付け保守による修理遅延の影響について検討を行っている[37]。しかし、ノード内で二重化し、さらに２つのノード間で相互バックアップを行ったシステムへの拡張や、システムの緊急度合いに応じて駆付け時間を二段階に分けた評価等は行っていない。一般に、二重化された単一ノードの場合は、経験的に、駆付けの緊急度合いはシステムダウンしているか否か等により判断される場合が多い。しかし、二重化された２つのノードが更に相互にバックアップされる場合には、駆付け要求が発生する状態が多く、どのようなときに緊急駆付けとするかの判断が重要となる。地理的に離れたノード間での相互バックアップは大規模災害への対応が本来の目的であるが、通常の装置障害に対しては、冗長度が上がるため、高い信頼性が容易に達成可能となる。許容される範囲での信頼性低下のもとで、駆付け時間の長時間化を図ることにより、保守の効率化とシステムの信頼性をバランス良く実現することが期待できる。

本論文では、データベースを持ち、ノード内は二重化されたモジュールが相互にバックアップし、さらに地理的に離れた地点に配置された２つのノードが相互にバックアップする高度ＩＮにおけるサービス制御ノードを対象として、保守の効率とシステム信頼性をバランスよく実現できる駆付け保守の方法を提案する。具体的には、システム信頼度の時間的な変化に着目して、駆付けの緊急度合いをシステム内の障害装置

数や、保守チームが到着しているか否かの状態によりレベル分けし、このレベルにより駆付け保守の方法を設定する。これに基づいて、システムの信頼性や駆付け回数を総合的に評価し、非緊急駆付けについては複数の障害をスケジュール化可能で、直ちに駆付けなければならない緊急駆付け回数の少ない効率的な保守方法を明らかにする。

1.5　論文の構成と各章の概要

本論文は、上記で述べたように高度インテリジェントネットワークにおけるサービス制御ノードのハードウェアシステム構成技術に関する研究成果をまとめたものであり、以下の8章から構成される。技術的課題と論文の構成との関係を図1.5に示す。

第1章では、本研究の背景と目的を示し、それに関連する研究状況、問題の所在を明らかにする。

第2章では、本研究で対象とするサービス制御ノードのシステム構成を検討する上で重要な、データベースの配備形態や、呼の接続処理、および複数のサービス制御ノードによる多種類のサービスの実現方法を示す。

第3章では、サービス制御ノードのデータベースに適した分散型メモリデータベース構成を提案する。サービス制御ノードのデータベースは容量が小さく、アクセス頻度が極めて高い特徴を有し、オンライントランザクション処理システムで一般に用いられている磁気ディスク装置にデータベースを格納する方法は適用できない。データベースを多数のモジュールの主メモリに分散して格納し、トランザクションを効率的に分散処理する方法を明らかにする。分散型メモリデータベース構成により、高性能で経済的なサービス制御ノードを実現できる。

第4章では、データベースの分散配置とカスタマ毎のトラヒックのばらつきに起因して起こるモジュール間の負荷の偏りの評価方法や、負荷を平準化してモジュールの

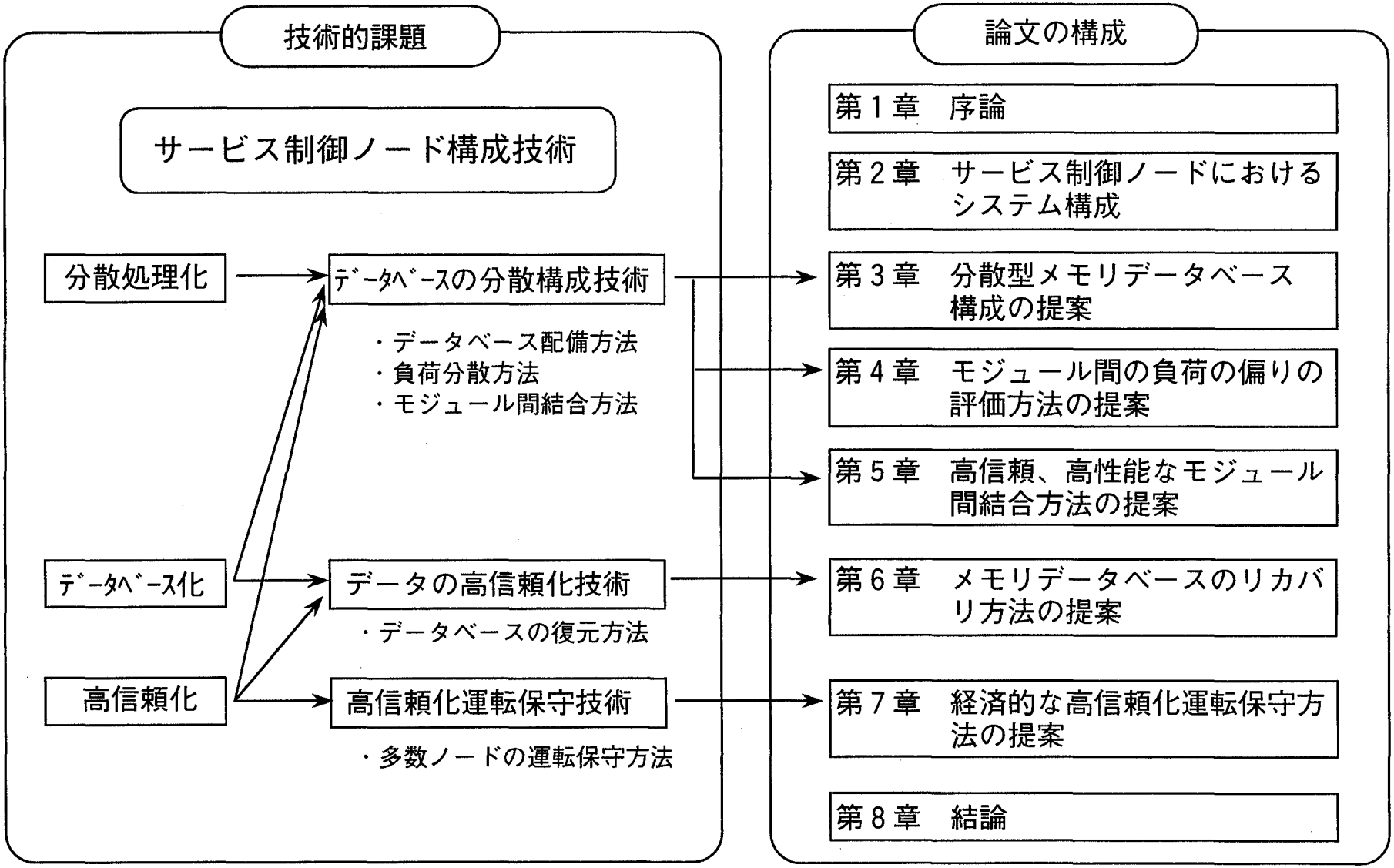


図１.５　技術的課題と論文の構成

稼働率を高める方法を提案する。カスタマ毎のトラヒック量については、トラヒックの高いものからの順位により近似する方法を示し、これを用いてモジュール間の負荷の偏りや、偏りの上限値の推定方法を明らかにする。また、負荷の平準化については、データの配置先を示すディレクトリの検索処理とデータベース処理を組合わせることにより、動的に平準化する方法を明らかにする。負荷の偏り評価や平準化技術は、分散処理によるサービス制御ノードの方式設計、設備設計に有効である。

第5章では、多数のモジュールによる分散構成を効率的に実現するため、高信頼で、拡張性の良いモジュール間の結合方法について提案する。インタフェースとしてはATMを適用し、結合機構を構成する機能ブロック毎に二重化することによりモジュール数が増加した場合でも高い信頼性が確保できる結合機構の構成法を明らかにする。また、モジュールの通信処理能力を向上させる通信処理方式を明らかにする。これにより、高信頼なサービス制御ノードを拡張性良く実現できる。

第6章では、主メモリ上に格納されたデータベースがソフトウェアのバグ、ハードウェア障害、電源障害により破壊、喪失された場合、データベースを効率的に回復する方法を提案する。呼の処理と並行して、データベースの更新ログと数分毎のチェックポイントデータベースを半導体ディスク装置に取得し、主メモリ上のデータベースが破壊、喪失した場合には、この半導体ディスク装置に取得した情報から障害直前のデータベースを高速に回復する。これにより、メモリデータベースの高信頼化を、通常の呼処理への影響が少なく実現できる。

第7章では、広域的に分散設置された多数のサービス制御ノードの効率的な運転保守方法を提案する。地震、火災等の大規模災害に対応するためには異なる地域に設置された2つのノード間での相互バックアップが必要であるが、これに伴うサービス制御ノードの信頼度の向上を運転保守の軽減に活用する方法を明らかにする。これにより、多数のサービス制御ノード群を効率的に運転保守することが可能となる。

第8章では、本研究で得られた結論について総括する。

第２章
サービス制御ノードにおけるシステム構成

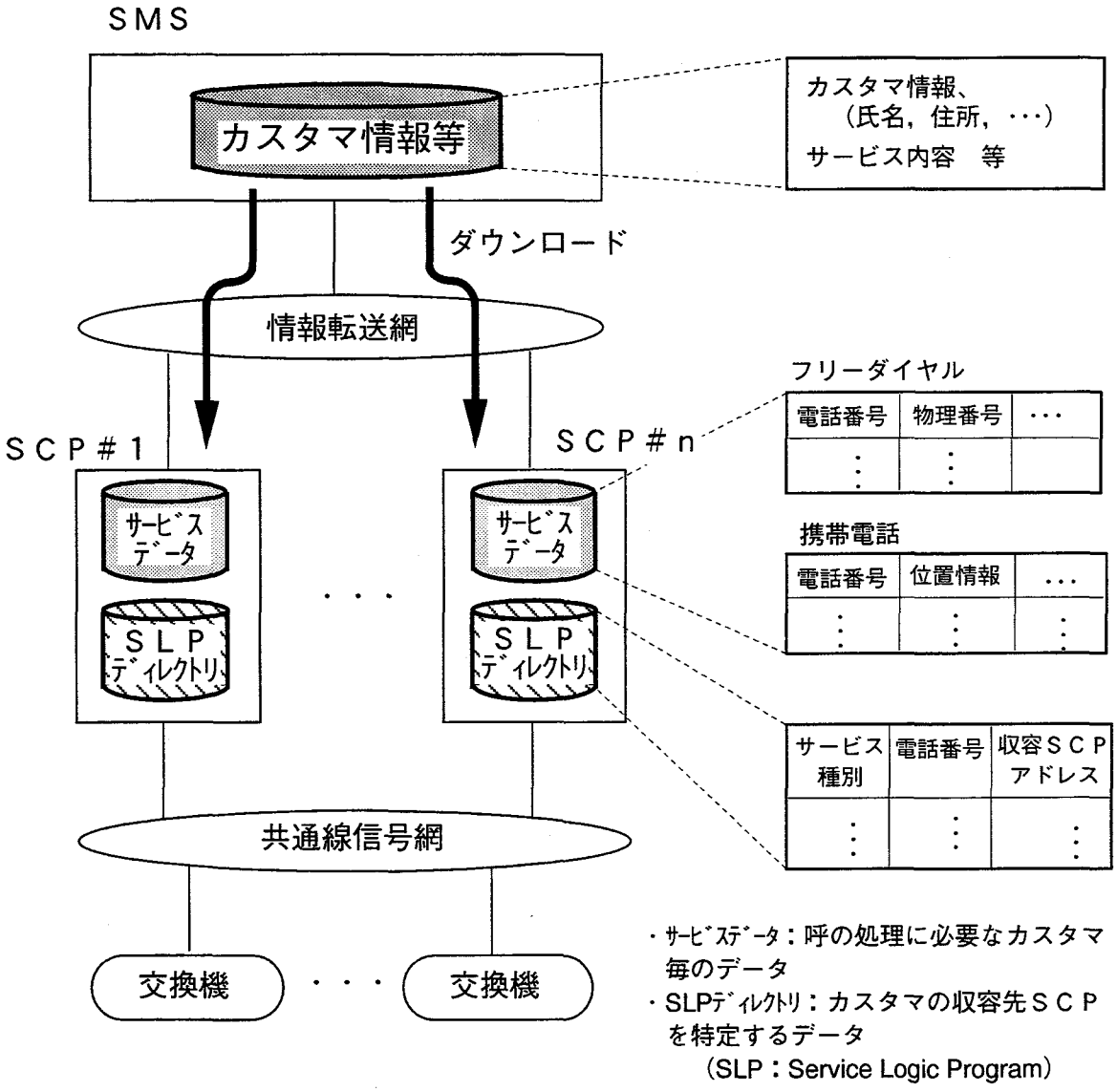
本章では、本研究で対象とするサービス制御ノードのシステム構成を考える上で重要な、データベースの配備、呼の接続処理、および今後のサービスの多様化に柔軟に対応可能とするための多種類のサービスの実現方法について述べる。

２.１　データベースの配備

高度インテリジェントネットワーク（高度ＩＮ）ではカスタマ毎に多様なデータベースを維持管理する必要がある。これらのデータベースはカスタマ管理に必要なデータベースと呼の処理に必要なデータベースに分類できる。前者は、カスタマの氏名や住所等のカスタマ情報やカスタマが受けるサービス条件等のデータベースからなる。一方、呼処理に必要なデータベースは電話番号や位置情報などからなる。高度ＩＮにおけるデータベースの配備形態を図２.１に示す[15]。

サービス管理ノード（ＳＭＳ）では、カスタマの情報や、そのカスタマが受けるサービス内容等がデータベースとして管理される。ここで管理される情報のうち、実際の呼処理に必要な情報のみが、サービス制御ノード（ＳＣＰ）にダウンロードされる。たとえば、フリーダイヤルの場合は、論理的な電話番号と物理番号の対などがデータベースとしてダウンロードされる。これらは、呼の接続サービスを行うために必要な情報であり、サービスデータと言われている。

ＳＣＰ内に保持されるデータベースとしては、サービスデータの他にＳＬＰ（Service Logic Program）ディレクトリがある。ＳＬＰはカスタマが受けるサービス（たとえば、フリーダイヤルの場合、時間帯により接続先の電話機を変更するとか、発信者の位置により接続する電話機を変更する等のサービス）を実現するため、カスタマ毎に作成されるプログラムである。このＳＬＰを書換えることにより、サービスのカ



・カスタマ管理に必要なデータベース（ＤＢ）はサービス管理ノード（ＳＭＳ）に配備

・呼処理に必要なデータベースはサービス制御ノード（ＳＣＰ）に配備

図２.１　高度インテリジェントネットワークにおけるデータベースの配備

スタマイズ化が容易に実現できる。ＳＬＰディレクトリとは、複数のＳＣＰにカスタマ毎のＳＬＰを分散収容した場合、どのカスタマのＳＬＰがどのＳＣＰに収容されているかを特定するためのデータベースである[5]。

2.2　呼の接続処理

電話機からの呼は一般の電話呼か高機能レイヤを利用する電話呼（以下、ＩＮ呼と記す）かを交換機で判別し、ＩＮ呼の場合は共通線信号網を介して処理要求がサービス制御ノードに送られる。サービス制御ノードはサービス毎、カスタマ毎にどのような内容のサービスを行うかのデータベースを持っており、この内容に従って交換機と協調して高機能なサービスを実現する。高度インテリジェントネットワークにおける呼処理の流れを図２.２に示す。

図２.２において、電話機から投入された電話番号は交換機で解読され、一般の電話呼かＩＮ呼かが判別される。一般の電話呼の場合は、投入された電話番号に基づいて接続処理が行われる（図２.２の破線の①と⑤）。ＩＮ呼の場合、たとえば最初の電話番号が０１２０（フリーダイヤル）や０９９０（ダイヤルＱ[2]）等で始まる呼の場合には、これらの電話番号の検出をトリガとして、あらかじめ決められたＳＣＰに問い合わせがかけられる。この時、トリガとなる電話番号に加えて、後続の電話番号がＳＣＰに送られる。ＳＣＰでは送られてきた電話番号をキーとしてデータベース処理を行い、論理番号から物理番号への変換を行う。フリーダイヤルの場合、カスタマによっては時間帯により接続先を変更したり、発信地点により着信先を振り分ける等の付加サービスを受けられている場合もあり、物理番号への変換に当たってはこのような条件も含めて行われる。次に、ＳＣＰはどのような処理を行うかの応答を交換機に返し、交換機はこの応答に基づいてＩＮ呼の接続処理を行う（図２.２の実線の①～⑤）。

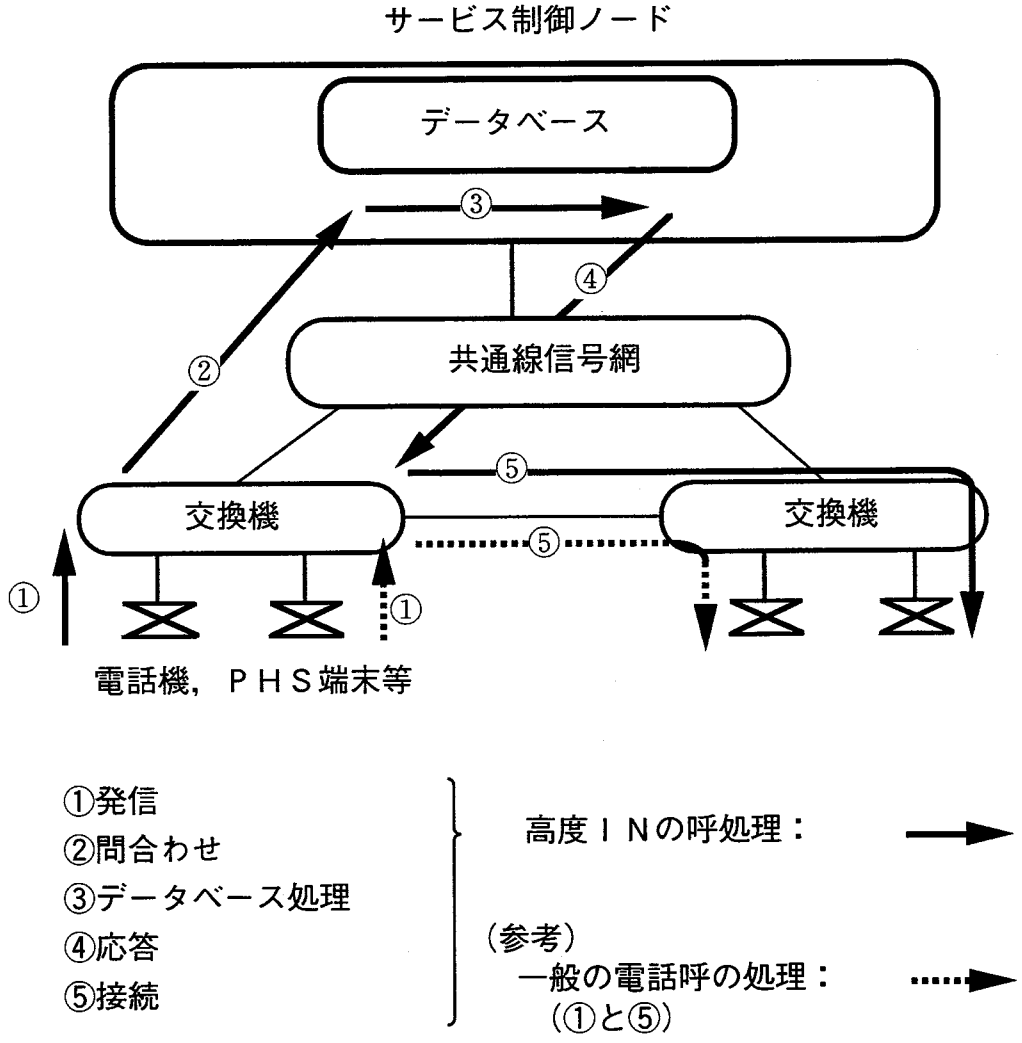
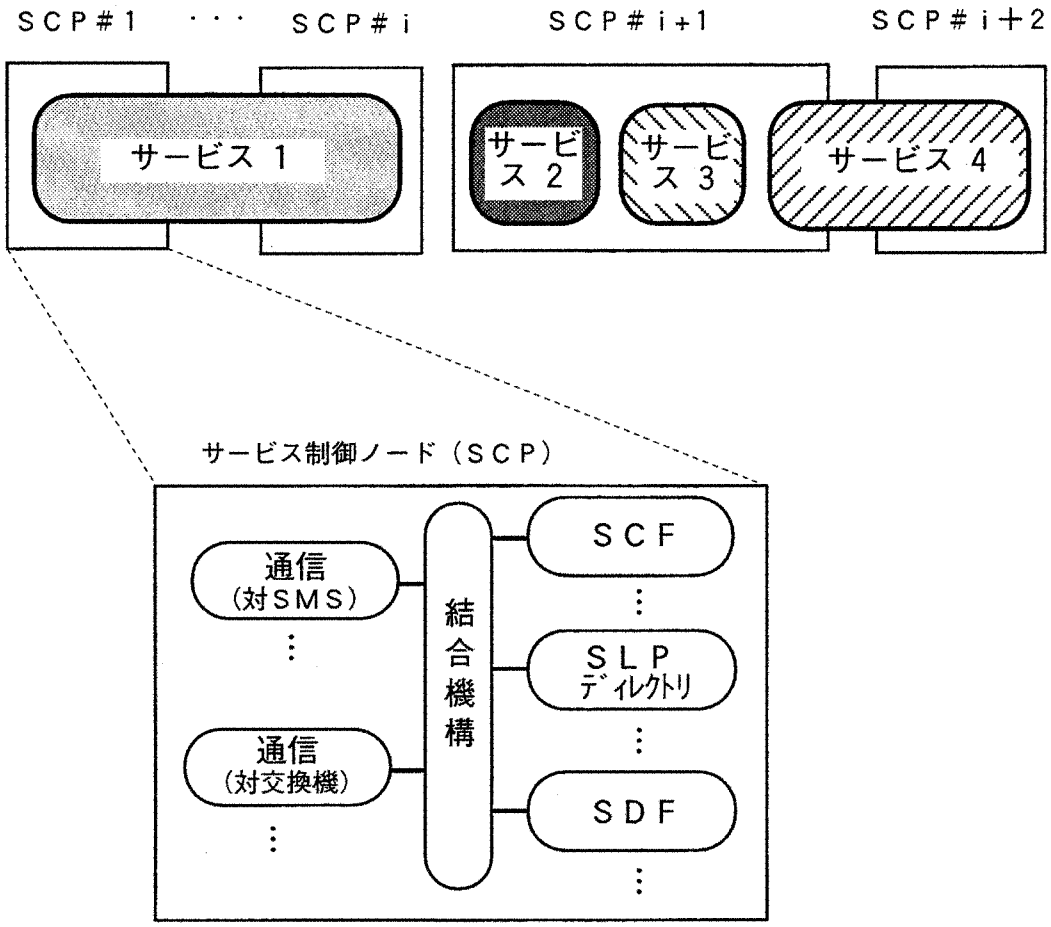


図２.２　高度インテリジェントネットワークにおける呼処理

2.3　多種類のサービスの実現方法

需要の異なる多種多様なサービスを効率的に実現するためには、大規模なサービスについては、複数のサービス制御ノードで分散することや、小規模なサービスについては、1つのサービス制御ノードで複数のサービスを実現することが必要となる。このためには、カスタマ毎のＳＬＰやサービスデータを複数のＳＣＰに分散収容し、カスタマ毎のトランザクションをＳＬＰやサービスデータの配備されているＳＣＰに割振ることが必要となる。トランザクションの振り分けは、前節で述べたＳＬＰディレクトリにより行われる。複数のサービス制御ノードによる多種類のサービスの実現方法を図2.3に示す。

また、サービス開始後の需要の増加や、サービス内容の高度化に伴ってプロセッサの処理量が増加した場合に、柔軟に対応するためには、サービス制御ノードを複数のモジュールで構成し、モジュールの追加により対応できるノード構成が必要となる[5]。サービス制御ノードの分散構成方式については次章以下で考察する。



多種類のサービスを需要に応じて柔軟に分散

・大規模サービス ⟶ 複数ノードで分散

・小規模サービス ⟶ 1ノードで複数サービス

・サービス開始後の需要の増加，サービス内容の高度化

⟶ サービス制御ノードを複数のモジュールで構成し，モジュールの追加により対応

図2.3　複数のサービス制御ノードによる多種類のサービスの実現方法

第３章
サービス制御ノードの分散構成技術

３．１　緒言

高度ＩＮのサービス制御ノード（ＳＣＰ）では、幅広いレンジにわたる多様な内容のサービスを、小規模から大規模まで、機能間の分離性を良好に保ちながら処理できること、多数のカスタマの情報から構成されるデータベースを効率よく高速に処理できることが重要となる。また、障害に対しても、これらデータベースの内容が十分な確度で保証されるとともに、サービスとしては高い可用性（アベイラビリティ）の達成が必須となる。

本章では、このような要求条件に効率的に対応可能な構成として分散構成に注目し、高度ＩＮのＳＣＰを多数のモジュールからなる大規模な分散構成により実現する方法を提案する[5] [38]～[40]。基本的な分散構成としては、ＩＴＵ－Ｔで標準化[41]されている、サービス制御機能（ＳＣＦ）とサービスデータ機能（ＳＤＦ）を別モジュールで実現し、さらにこれら機能の負荷を複数のモジュールで処理可能とした機能分散、負荷分散の統合形態としている。分散構成の実現に当たっては、高度ＩＮに適したデータベースの分散配備の方法を明らかにすることが主要な課題となる。はじめに、小容量でアクセス頻度が極めて高いこと等、高度ＩＮ用データベースの特性を明らかにし、それに基づいて、複数のモジュールからのデータベースの共用の可否、主記憶／半導体ディスク装置いずれかの格納媒体の選択の２点に注目して構成案を設定した。これら案について、モジュール間の負荷の偏り、データベース処理でのダイナミックステップ数に対する評価を加え、コスト、性能面から総合的に比較を行った。結果を高度ＩＮに適した分散型メモリデータベース構成として提案する。データベースのリカバリについては、バンキングシステム等のＯＬＴＰ（On-Line Transaction Processing）系システムで、データ内容の厳密な復元を重

要視することとは対照的に、通信システムとしての高度ＩＮでは、短いリカバリ時間の実現が最も重要となる。これを考慮し、データ毎の特性に応じたリカバリレベルを設定するとともに、数分毎のチェックポイントデータベースの取得と小容量の更新ログデータの組み合わせによる、効率的なリカバリ方法を明らかにする。また、一般に、分散構成下では、冗長化による高いアベイラビリティの実現は容易であるが、同時にコストアップを招きやすい。ノード内のモジュール間バックアップとノード間にわたるバックアップを効率的に組み合わせ、高いアベイラビリティを経済的に実現する冗長構成法を提案する。

３.２　ノード構成条件と機能配備

３.２.１　ノード構成条件

本章では、パーソナル通信など全国規模のサービス、カスタマオリエンティッドな多様なサービスへの適用を考え、従来ノード[10]に比べ大規模化し、１００万カスタマ程度を収容可能なノードを想定する。また、ノードにかかるトラヒックは２０００呼／秒程度を想定する。

３.２.２　機能配備

高度ＩＮに適用するＳＣＰを実用的なシステムとして構成するためには、以下の機能が必要である。

（１）**サービス制御機能（ＳＣＦ）**：カスタマ毎に作成されたサービスロジックプログラム（ＳＬＰ）によりサービスを実行する機能。

（2）**サービスデータ機能（ＳＤＦ）**：サービスを実行するために必要なサービスデータを保持し、ＳＬＰからの問い合わせに従ってサービスデータへのアクセスを行う機能。

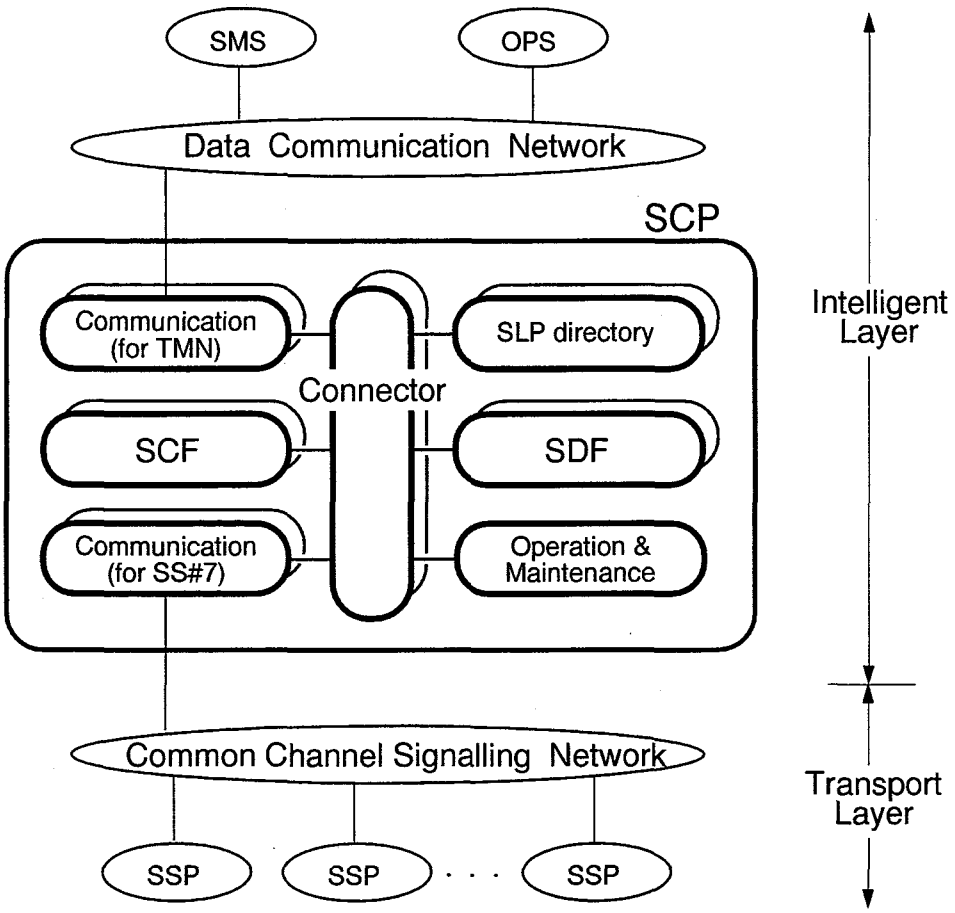
（3）**ＳＬＰディレクトリ検索機能**：ネットワーク内に複数のＳＣＰを配備し、ＳＬＰを分散収容した場合、ＳＬＰがどのＳＣＰに配備されているかを決定する機能。

（4）**他ノードとの通信機能**：共通線信号網を介した交換機との通信、パケット網を介したサービス管理システム（ＳＭＳ）との通信機能。

（5）**運転・保守機能**：上位ノードからの指示に基づく運転操作やノード全体の障害管理を行う機能。

これら（1）から（5）の機能のモジュールへの配備に関しては、機能間の分離性がよく、負荷変動に対して機能単位に柔軟に対応できること、実現する機能に最適化した技術を適用できることから、機能毎に別々のモジュールに配備し、負荷の増大に対しては機能単位で負荷分散することとした。共通線信号網及びパケット網を介した他ノードとの通信機能については、各網の将来の発展形態が異なることから、それぞれ別モジュールで分担することとした。ＳＣＰのネットワーク内の位置付け及びＳＣＰ内の機能配備を図3.1に示す。

なお、本章では、ＳＬＰディレクトリはＳＣＰに配備した。交換機側はカスタマによらず、前もって定めたＳＣＰにアクセスする。そのＳＣＰでＳＬＰディレクトリを検索し、当該カスタマのＳＬＰが配備されているＳＣＰを知る。その結果に基づき、呼を当該ＳＣＰへ転送する。本方式のほかに、交換機側にＳＬＰディレクトリを配備する方法も考えられるが、ＳＬＰの追加毎に数１０００台に及ぶ交換機上でのＳＬＰディレクトリの変更が必要となる。ネットワーク管理上のオーバヘッド



SMS: Service Management System
OPS: Operations System
SSP : Service Switching Point
SCF : Service Logic Function
SDF : Service Data Function
SLP : Service Logic Program

図３．１　ＳＣＰの機能配備

が極めて大きいため対象外とした。

ＳＣＦの負荷分散については、ＳＣＦを分担する全てのモジュールにＳＬＰを重複して配備した。これにより、ＳＣＦを分担するどのモジュールでもＳＬＰを実行可能とし、特定のカスタマに負荷が集中した場合においても均等な負荷分散を可能とした。他ノードとの通信機能の負荷分散については、モジュール内の回線増設とモジュール増設を組み合わせて対応することとした。ＳＤＦ、ＳＬＰディレクトリ検索のデータベース処理の負荷分散については、負荷の偏りや、どのような記憶媒体にデータベースを格納するか等の問題を併せて検討するため、次節で考察する。なお、運転・保守機能については、ノードが大規模化されても、１台のモジュールで対応可能との見通しを得たため、負荷分散は行わないものとした。

３.３　データベースの分散構成

高度ＩＮにおけるデータベースの特性について考察し、次にその特性にあった複数モジュールへのデータベースの配備、分散方法を明らかにする。

３.３.１ サービス制御ノードにおけるデータベースの特性

ＳＬＰディレクトリとサービスデータから構成されるデータベースの容量、アクセス特性等を表３.１に示す。各データベースの主要な特徴は以下の通りである。

（１）ノード当たりのデータベース容量は数百ＭＢから１ＧＢであり、ＯＬＴＰ系システム、たとえばバンキングシステムで必要とされるデータベース容量に比べ、おおよそ１／１０～１／１００と小さい。一方、単位時間、単位容量当たりのＩ／Ｏアクセス回数（以下、Ｉ／Ｏアクセス密度と記す）は１００～１００００回／秒／ＧＢであり、ＯＬＴＰ系システムに比べて１００

表３.１　サービス制御ノードにおけるデータベースの特性

	用途	データベース容量	データベースアクセス頻度	入出力アクセス頻度	入出力アクセス密度
ＳＬＰディレクトリ（参照系データ）	収容先ＳＣＰを特定	～1GB	２k回／s	４k回/ s	10 k回/s/GB
サービスデータ（更新系データ）	呼処理	～1GB	5～10k回/s	10～20k回/s	10～20 k回/s/GB

（注）・条件：ノード当たりのトラヒックは２０００呼／秒を想定
（１００万カスタマ収容時）
・入出力アクセス頻度：単位時間当たりの入出力アクセス回数
（データベースが外部記憶装置に格納されている場合）
・磁気ディスク装置の入出力アクセス密度＝数10回/s/GB

～１０００倍大きい。すなわち、高度ＩＮ用データベースでは少ない容量のデータベースに極めて高頻度のアクセスがある。

（２）高度ＩＮのカスタマ毎のレコード（データベースの最小操作単位）への単位時間当たりのアクセス回数（以下、アクセス頻度と記す）はサービス種別にもよるが、一般的に大きくばらつく。少数のカスタマが高いアクセス頻度を有し、かつ全体のアクセス頻度の相当程度を占める。アクセス頻度の高いカスタマとアクセス頻度の低いカスタマのアクセス頻度の比は１０００倍から１００００倍と大きい。

（３）データベースへのアクセスは、電話機から入力されたダイヤル番号（フリーダイヤルサービスの場合、「０１２０」の後に続く６桁の論理番号）等をユニークキー（データベースで、あるデータ項目に着目した時、同一の内容が存在せず、カスタマ毎のレコードを一意に認識できる項目）とし、そのキーに一致する１レコードをアクセスする形態がほとんどである。

（４）データベースへのアクセスには、呼処理のアクセス（交換機からのアクセス）とサービス管理のアクセス（新規カスタマの追加、サービス条件等の変更に伴うＳＭＳからのアクセス）がある。サービス管理のアクセスでは情報がＳＭＳよりダウンロードされデータベースの書き換えが行われる。呼処理のアクセスでは、ＳＬＰディレクトリに対しては読み出しのみが行われ、サービスデータに対してはサービス内容に応じて読み出し／書き込みが行われる。呼処理のアクセスに比べ、サービス管理のアクセスは２～３桁アクセス頻度が少なく、各データベースへのアクセス特性としては呼処理のアクセスのみを考えれば十分である。従って、ＳＬＰディレクトリは読み出し専用データベース（以下、参照系データベースと記す）、サービスデータは読み出し／書き込みデータベース（以下、更新系データベースと記す）として扱う。

3.3.2　データベースの格納媒体

データベース格納媒体として、磁気ディスク装置、半導体ディスク装置、モジュール内プロセッサの主記憶（以下、メモリと記す）がある。本章では、以下の理由により、メモリと半導体ディスク装置を対象とした。

（1）一般にＯＬＴＰ系システム[20][21]に適用されている磁気ディスク装置、半導体ディスク装置の単位時間当たり処理可能なＩ／Ｏ回数はそれぞれ４０～６０回／秒、１０００～２０００回／秒である。高度ＩＮでのノード当たり要求能力が１００００～２００００回／秒（表3.1）であるため、最大１ＧＢ程度のデータベースを格納するために、磁気ディスク装置は数１００台を並べる必要があり適用性はない。一方、半導体ディスク装置は１０台程度にＩ／Ｏアクセスを分散させることにより適用の可能性がある。

（2）データベースの容量が比較的小さく、アクセス頻度の高い分野のデータベースでは、データベース全体をメモリに常駐させるメモリデータベースが有効であると言われている[28][29]。高度ＩＮのデータベースは最大でも１ＧＢ程度と容量が小さく、アクセス頻度が非常に高いためメモリデータベースの適用性がある。ただし、ハード、ソフト障害時にデータベースが破壊されやすいため、これに対応可能なリカバリ方式を併せて採用する必要がある。

3.3.3　データベース分散方法

3.3.3.1　複数モジュールへのデータベースの配備法

複数モジュールからのデータベース（ＤＢ）の共用の可否、格納媒体の違い（メモリ、半導体ディスク装置（ＳＤ））に注目し、以下の３方式を構成上の代替案と

して比較評価する。

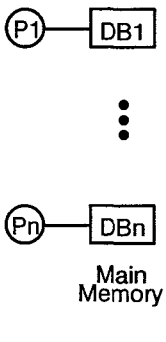
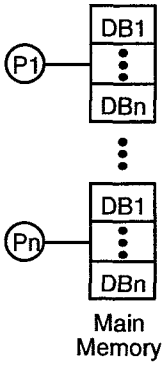
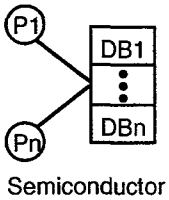
【分割方式（Partitioned Method）：ＤＢ非共用／メモリ格納】データベースをレコード単位にモジュール台数分に分割し、各モジュールのメモリに格納する方式。カスタマ毎のレコードへのアクセス頻度の偏りに起因してモジュール間の負荷の偏りが生じやすい。また、高度ＩＮのデータベース処理はユニークキーによる１レコードアクセスがほとんどで、モジュール間に跨った処理は極めて少ない。このため、本方式のようにデータベースを分割配備したことによるデータベース処理のオーバヘッドは通常無視できる。しかし、データベースアクセスに先だって、どのレコードがどのモジュールに格納されているかを判別するための処理が必要となる。

【多重方式（Duplicated Method）：ＤＢ共用／メモリ格納】全てのデータベースを各モジュールのメモリに重複して格納する方式。メモリを重複して持つため、データベース格納コストが他の方式に比べて高い。更新系データベースに適用した場合、データベースの一致を保証するため、あるモジュールがデータベースを書き換えると、他の全てのモジュールのデータベースも書き換える必要がある。また、データベースを共用するため排他制御が必要となる。

【ＳＤ共用方式（Shared Method）：ＤＢ共用／ＳＤ格納】半導体ディスク装置に全てのデータベースを格納し、全てのモジュールで共用する方式。データベースが半導体ディスク装置上にあるため、入出力処理のオーバヘッドが生ずる。また、更新系データベースに適用した場合は、多重方式と同様に排他制御が必要となる。

各方式の構成を表３．２に示す。なお、データベースをレコード単位にモジュール台数分に分割し、各モジュールの半導体ディスク装置に格納するＳＤ非共用方式

表３.２　データベースの分散構成

略称		分割方式		多重方式		ＳＤ共用方式	
データベースの共用の可/否		非共用		共用		共用	
データベースの格納媒体		メモリ		メモリ		半導体ディスク装置(ＳＤ)	
構成 Ｐ：プロセッサ ＤＢ：データベース							
モジュール間の負荷の偏り		△	有り	○	無し	○	無し
プロセッサの処理量 (ﾀﾞｲﾅﾐｯｸ ｽﾃｯﾌﾟ数)	モジュール選択	△	要	○	不要	○	不要
	DB一致処理	○	不要	×	要	○	不要
	排他制御	○	不要	△	要	△	要
	入出力処理	○	無し	○	無し	△	有り
データベースの格納コスト		○	安い	△	高い	○	安い

（ＤＢ非共用／ＳＤ格納）は、ＤＢ非共用／メモリ格納の分割方式に比べ、入出力オーバヘッドの発生などで明らかにコスト性能比が劣るため除外した。

これら３方式について、まず分散構成でのモジュール間の負荷の偏り、ダイナミックステップ数について考察する。次に、これら要因と性能とを総合し、システムコストとして比較評価を行う。本章では、ノードのスループットを一定とし、レスポンスタイムは一定値以下の条件でのコスト評価を行う。スループットはモジュール台数を通じて、レスポンスタイムはモジュールのプロセッサ使用率を通じてコストに反映される。

３.３.３.２　モジュール間の負荷の偏り

多重方式、ＳＤ共用方式では、データベースを共用しており、どのモジュールからも任意のデータベースへのアクセスが可能である。ＳＣＦからラウンドロビン方式により各モジュールへデータベースアクセスを割り振ることにより、短時間でのばらつきは生ずるとしても、各モジュールへのアクセス頻度は一様となる。

一方、分割方式では、データベースアクセスはカスタマのレコードを格納しているモジュールでのみ処理可能である。このため、モジュール間のアクセス頻度に偏りが生じやすい。

アクセス頻度の最も高いモジュールのアクセス頻度の平均値（λ_0）からの偏りの割合を負荷偏り率（ω）とおく。負荷偏り率ωについては、第４章で詳細に述べるが、カスタマ毎のアクセス頻度が１万倍程度と大きくばらついたとしても、モジュールに収容するカスタマ数が数万程度になれば、負荷偏り率ωは０.２程度以内に収まる。

負荷の最も高いモジュールのアクセス頻度は（１＋ω）λ_0で表せる。このモジュールにおいても、要求されるレスポンスタイムを保証するためには、あらかじめ偏りを考慮して平均アクセス頻度の（１＋ω）倍の負荷を見込んだ設備設計をする必要がある。

3.3.3.3　ダイナミックステップ数の評価

分散構成を採ったときの1呼当たりのダイナミックステップ数（以下、ＤＳと記す）を表3.3に示す要因に従い算出した。特に、表3.3中のＩ／Ｏオーバヘッド（D_I）とは、データベースが半導体ディスク装置上に格納されていることによって生ずるオーバヘッドであり、データベースを半導体ディスク装置上に格納した場合とメモリ上に格納した場合のＤＳの差を示す。具体的には、半導体ディスク装置とメモリとの間の入出力処理のＤＳはもちろんであるが、さらに入出力処理のためにデータベースをページ単位で管理するためのＤＳを含む。従来、データベースを外部記憶装置に格納する場合は、入出力処理の容易化のためにデータベースをページ単位で管理してきた。しかし、メモリデータベースの場合は、管理単位をページにとらわれることなく構成することが可能となる。これによりＤＳの少ないデータベースアクセスが実現できる。図3.2に示すように、データベース処理のＤＳを約1／2に削減できるとの結果を得た。なお、図中のＤＳには、分散構成を採ることにより生ずるモジュール選択処理、排他制御等のオーバヘッドは含めていない。以上のことから、Ｉ／Ｏオーバヘッドは表3.3のD_0と同程度とした。

参照系データベース、更新系データベースについてＤＳの比較結果を図3.3、図3.4に示す。図は1呼当たりのデータベースアクセス回数（Ｑ）が変化した場合のＤＳを示している。図3.3より、参照系データベースでは、データベース一致処理、排他制御が不要であるため、多重方式のＤＳが最も少ない。一方、更新系データベースでは、図3.4より、データベース一致処理のＤＳがモジュール台数Ｎに比例して大きくなるため、多重方式のＤＳが最も大きい。分割方式とＳＤ共用方式を比べた場合、分割方式はメモリデータベースのため、半導体ディスク装置格納のデータベースであるＳＤ共用方式に比べてＤＳが少ない。

表3.3　分散構成をとったときのDSの要因

要　因	内　容
データベース処理（D_0）	・データベースの格納構成に依存せず各方式に共通にかかるＤＳ ・データベースアクセス回数に比例
モジュール選択処理（D_S）	・分割方式で、どのレコードがどのモジュールに格納されているかの判別のために必要なＤＳ ・全てのモジュールがレコードの配備先を示すディレクトリを持ちＳＣＦからのデータベースアクセスに対しては、まず当該ディレクトリを検索し、この結果に従ってレコードを格納しているモジュールにアクセスする方式を採用
データベース一致処理（D_W）	・多重方式で、かつ更新系データベースの場合、あるモジュールがデータベースを更新するたびに他の全てのモジュール（ｎ－１）台のデータベースを更新するために必要なＤＳ
排他制御（D_L）	・多重方式及びＳＤ共用方式でかつ更新系データベースの場合、複数のモジュールがデータベースを共用するための排他制御に必要なＤＳ
I/Oオーバーヘッド（D_I）	・ＳＤ共用方式の場合、データベースが半導体ディスク装置に格納されており、半導体ディスク装置とメモリとの間の入出力処理及び入出力が容易なページ単位でのデータ管理処理のために必要なＤＳ

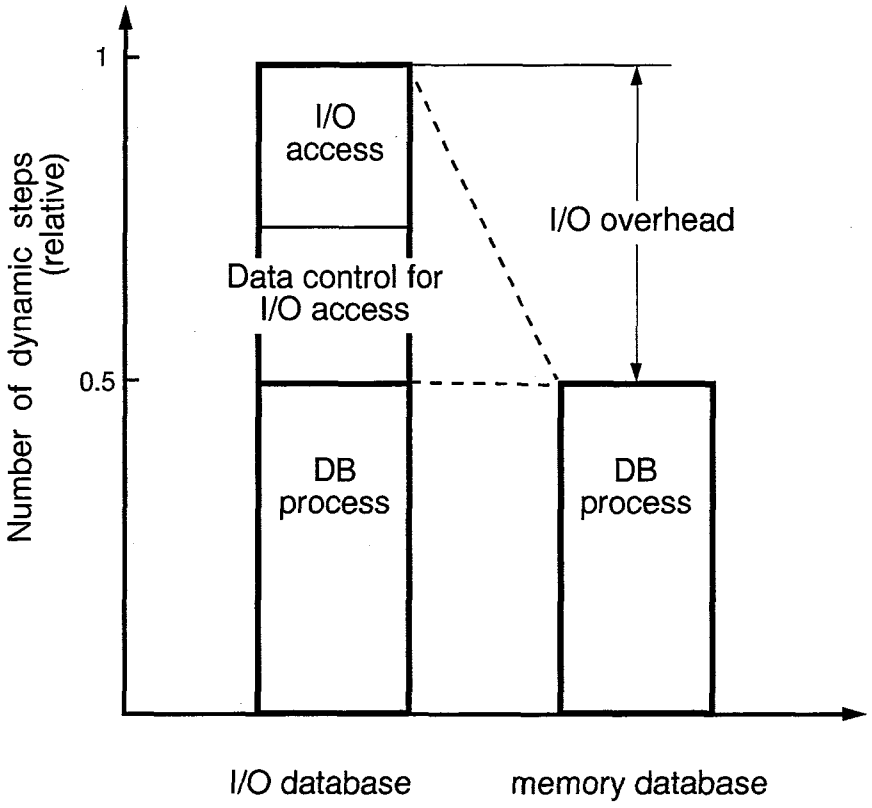


図３.２　メモリデータベースの効果

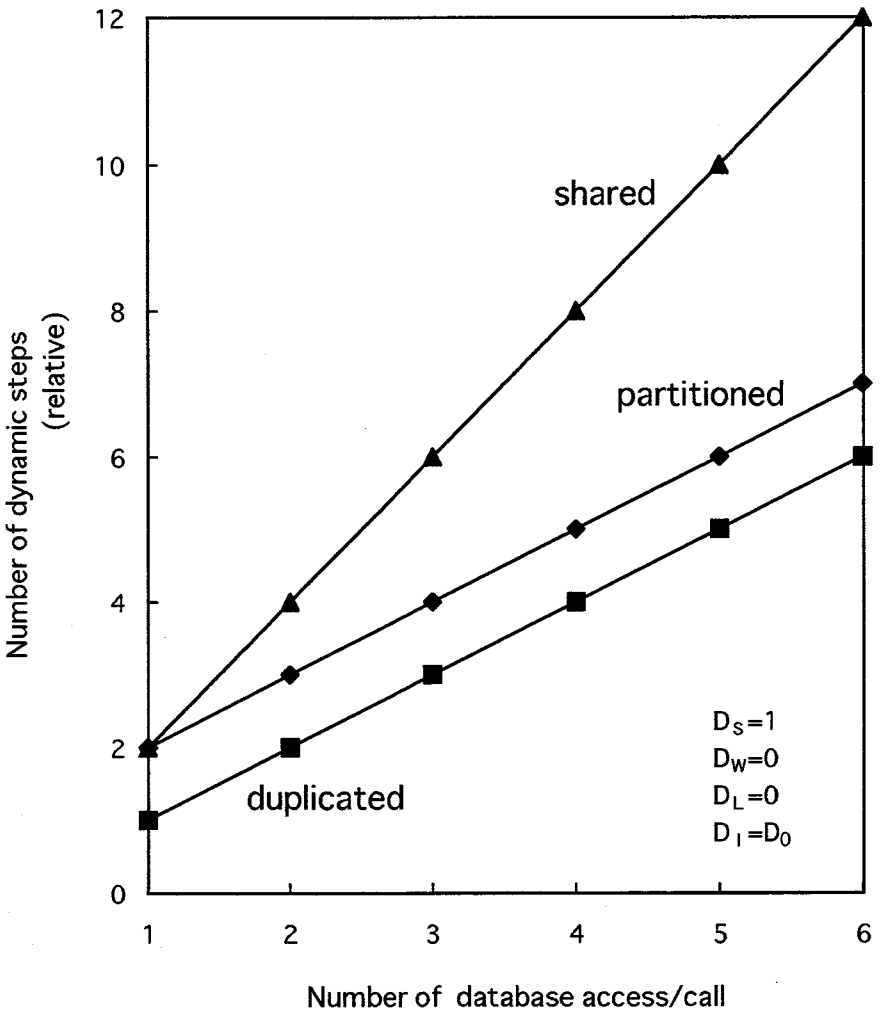


図３.３　ＤＳの比較（参照系データベース）

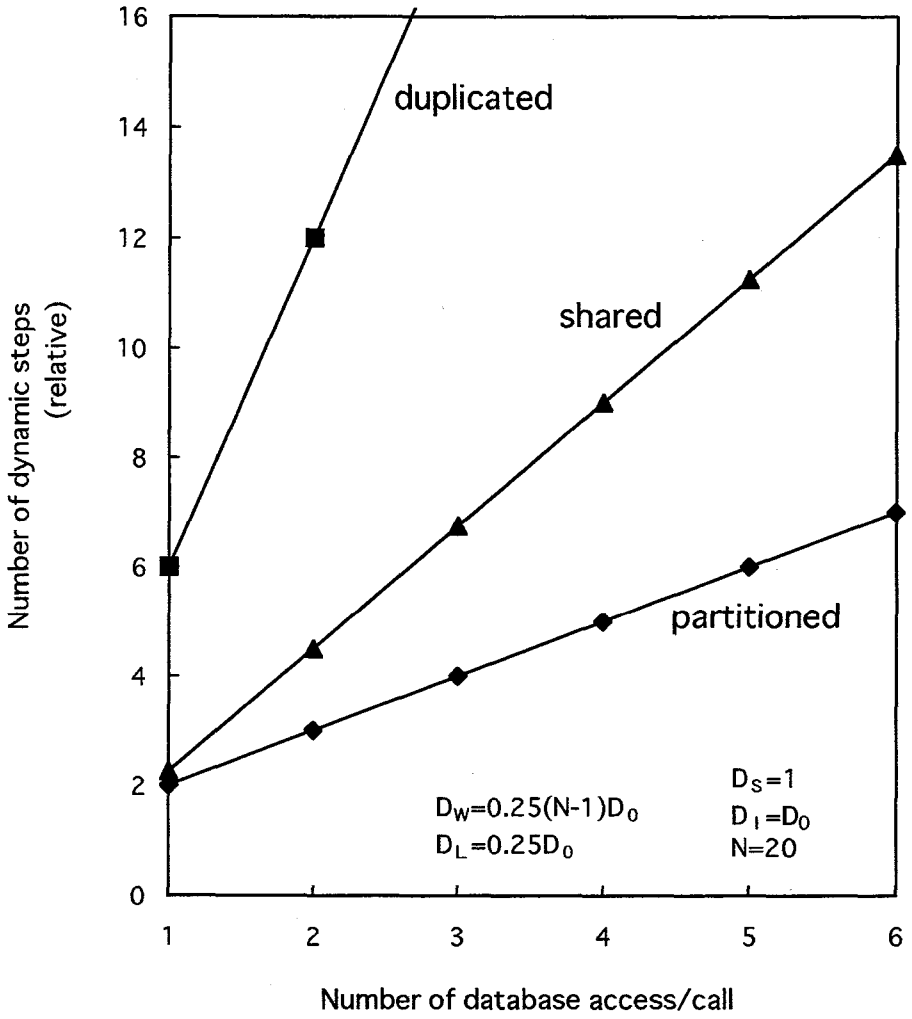


図３．４　ＤＳの比較（更新系データベース）

3.3.3.4　コスト評価

各方式の優劣をＤＳの大小、ハードウェア性能を考慮し、総合的に比較するため、スループットを同一として、各方式でのノードコストを比較する。ここで、ノードコストとは、ノード実現に必要なハードウェアコストとした。分割方式、多重方式、ＳＤ共用方式のノードコストをそれぞれF_{PA}、F_{DU}、F_{SH}また、ＤＳをD_{PA}、D_{DU}、D_{SH}としたとき、ノードコストを以下の式で表す。

$$F_{PA} = \lambda k_{CPU} D_{PA}(1+\omega\eta)/\rho + k_{MEM} M \tag{3.1}$$

$$F_{DU} = \lambda k_{CPU} D_{DU}/\rho + k_{MEM} M \frac{\lambda D_{DU}}{\rho P} \tag{3.2}$$

$$F_{SH} = \lambda k_{CPU} D_{SH}/\rho + k_{SF} M \tag{3.3}$$

Ｐ：プロセッサ性能

ρ：プロセッサ使用率

Ｍ：データベース容量

λ：ノード全体にかかるトラヒック

k_{CPU}：命令当たりのプロセッサコスト

k_{MEM}：単位容量当たりのメモリコスト

k_{SF}：単位容量当たりの半導体ディスク装置のコスト

ここで、D_{PA}は表３.３に示したD_0とD_Sの和で表せる。D_0はモジュール間で負荷の偏りが生ずるが、D_Sはどのモジュールでも処理可能であるため、負荷の偏りは生じない。ηはD_{PA}のうち、負荷の偏りが生ずるＤＳの割合（$\eta = D_0 / D_{PA}$）である。分割方式については、負荷の偏りを次式によりモジュール台数増に反映した。

$$N_{PA} = \frac{\lambda(1+\omega)D_{PA}\eta + \lambda D_{PA}(1-\eta)}{\rho P}$$

$$= \frac{\lambda D_{PA}(1+\omega\eta)}{\rho P} \tag{3.4}$$

式（3.1）～（3.3）の第１項はプロセッサコストであり、第２項はデータベース格納コストである。スループットを同一としているため、F_{PA}、F_{DU}、F_{SH}で表されるノードコストが各方式のコスト／性能を表す。

参照系データベース、更新系データベースについて、データベース格納構成のコスト比較結果を図３.５、図３.６に示す。プロセッサ、メモリのコスト、性能データは、現時点での市販品を参考に設定した。

図３.５より、データベース容量が数ＧＢ以下では、プロセッサコストに比べデータベース格納のためのコストは小さく、多重方式が最も有利である。従って、参照系データベースのＳＬＰディレクトリでは多重方式をとることとした。これに対し、更新系データベースの場合は、図３.６より、分割方式とＳＤ共用方式の優劣は、分割方式の負荷偏り率ωに依存する。前節で述べたようにωは０.２程度である。ωがこの程度であれば分割方式が有利である。従って、更新系データベースのサービスデータについては分割方式をとることとした。

３.４　分散処理ノードの高信頼化

本節では、データベースのリカバリ方法と分散構成の特質を生かした冗長化構成法について明らかにする。

３.４.１　データベースのリカバリ方法

ハード／ソフト障害によりメモリ上のデータベースは、外部記憶装置格納のデータベースに比べて、破壊されやすい。メモリ上のデータベースが破壊された場合のリカバリ方法として、ＯＬＴＰ系システムで一般に用いられている方法は以下の通りである[20]。

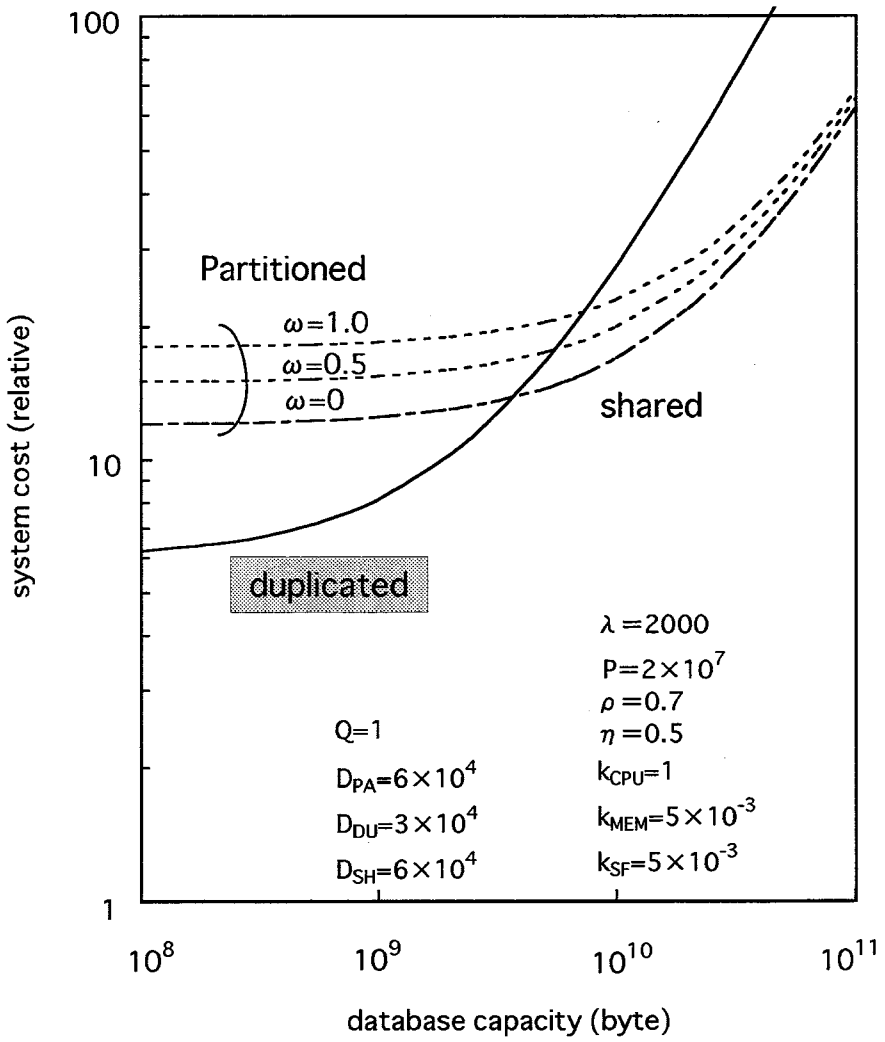


図３.５　コスト比較（参照系データベース）

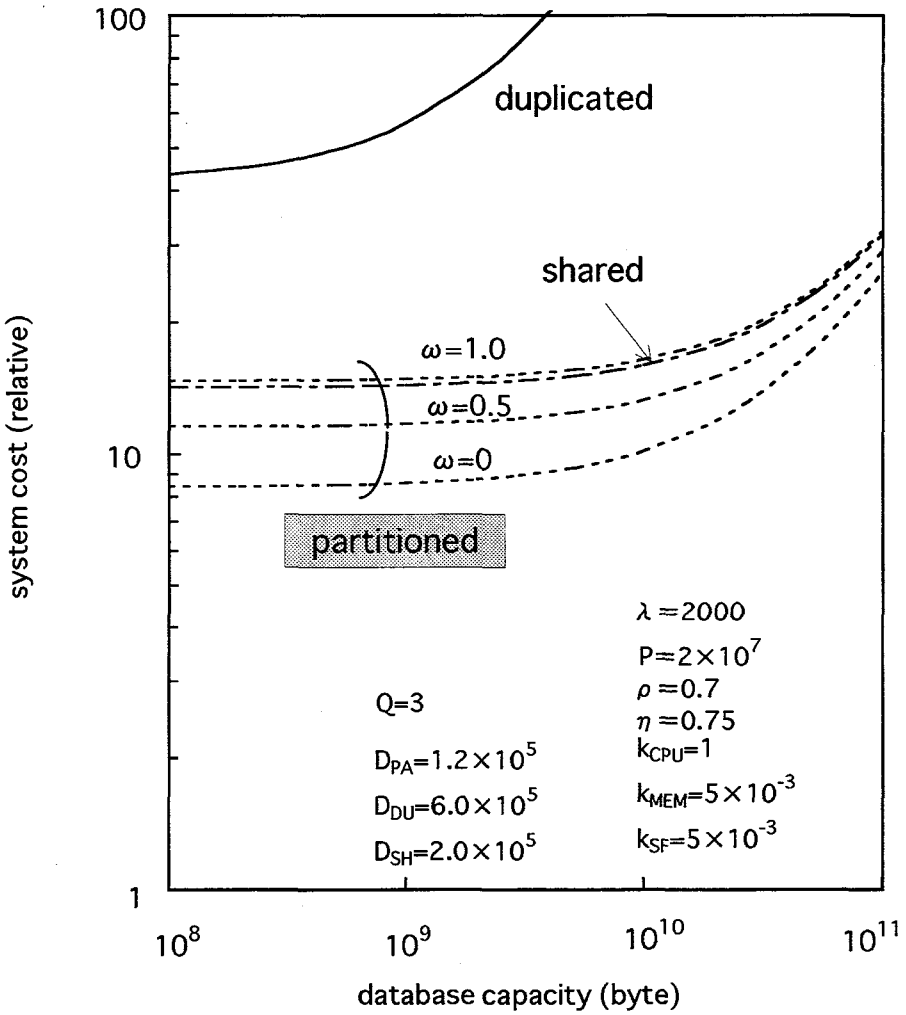


図３．６　コスト比較（更新系データベース）

（1）外部記憶装置（磁気ディスク装置又は半導体ディスク装置）にチェックポイント（ＣＰ）時点のデータベース（ＣＰＤＢ）とチェックポント以降のデータベースの更新ログ（ＬＯＧ）を格納しておく。

（2）障害検出時には外部記憶装置からＣＰＤＢとＬＯＧを読み出し、ＣＰＤＢにＬＯＧを上書きすることにより障害直前のデータベースを復元する。

しかし、バンキングシステムをはじめとするＯＬＴＰ系システムと高度ＩＮとではデータベースのリカバリに関する条件は大きく異なる。バンキングシステムでは扱うデータは金銭のような重要なデータがほとんどであり、障害やバグの種類を問わずデータベースの復元が必須とされる。通常のハード／ソフト障害に対して、データベースを完全に復元することはもとより、発生頻度は極めて少ないが、アプリケーションプログラム（以下、ＡＰＬと記す）等の論理バグに起因するデータベース破壊に対しても対処が必要となる。このため、長時間に渡ってＬＯＧを取得しておき、リカバリ時間は長くなるとしても、厳密に復元することを目指している。

一方、高度ＩＮで扱うデータには、パーソナル通信の位置情報のようにデータを喪失した場合、サービスに直接影響するデータと、回線が使用中か否かを表示するフラグのようにリカバリ時に初期値に戻せばサービスを行う上で支障が生じないデータがある。さらに、位置情報はカスタマの移動とともに更新されるため、リカバリに時間がかかると、たとえデータベースを復元したとしても実際の位置との不一致が生ずる。また、内容が一部喪失したとしても、位置情報の管理エリアをカスタマが移動した場合などには、再登録が行われ、それ以降は支障無くサービスが継続できる。このため、データの重要度に応じたデータベースのリカバリが要求されるとともに、データベースの完全な復元よりもサービス再開を迅速に行うことが優先される。

西原らは、情報の重要度に応じてデータベースの復元レベルを、①初期値に戻す（レベル０）、②ＣＰ値に戻す（レベル０.５）、③最新値に戻す（レベル１）に分

類し、ＬＯＧ取得をレベル１の情報のみに絞る提案を行っている[42]。ここでは、上述した高度ＩＮのリカバリに対する要求条件の分析を通して、レベル１の情報に対し、さらに復元の厳密さとリカバリ時間の２点から考察を加え、表３.４に示すように情報の重要度を新たに設定した。ここで、復元の厳密さとリカバリ時間に関しては、ＬＯＧ取得時間すなわちＣＰから障害検出までの時間（T_{CP}）と、障害発生から障害検出までの時間（T_B）に着目して考察した。これらの時間関係を図３.７に示す。

一般に、ハード／ソフト障害には、発生後数分以内に検出される障害（以下、障害レベルＡと記す）と検出までに数分以上かかる障害（以下、障害レベルＢと記す）がある。障害レベルＡはハードの２重化チェック、パリティチェックなどで検出されるハード障害、ウォッチドックタイマ等で検出されるソフト暴走等の障害で、通常のほとんどの障害がこれに該当する。障害レベルＢは発生頻度は極めてまれであるが、ＡＰＬ等の論理バグに起因する障害、ハードの多重障害のためチェック機構により検出できなかった障害であり、利用者やＡＰＬレベルでの矛盾発生により検出される障害である。障害発生以降も障害検出まではトランザクション処理は継続され、ＬＯＧは取得される。障害発生以降はデータベースの一部が破壊されている可能性がある。しかし、いつ破壊されているかは不明であり、障害発生から障害検出までを対象にリカバリを行う必要がある。リカバリを確実に行うためには、ＣＰＤＢが保証されていることが必要であり、$T_{CP} > T_B$を満足するＣＰＤＢからのリカバリが必要となる。障害レベルＢからのデータベースの復元を厳密に行うためには、T_{CP}の長いＣＰＤＢから復元することとなり、リカバリに長い時間がかかる。

以上より、以下に示す方法によりデータベースのリカバリを行うこととした。

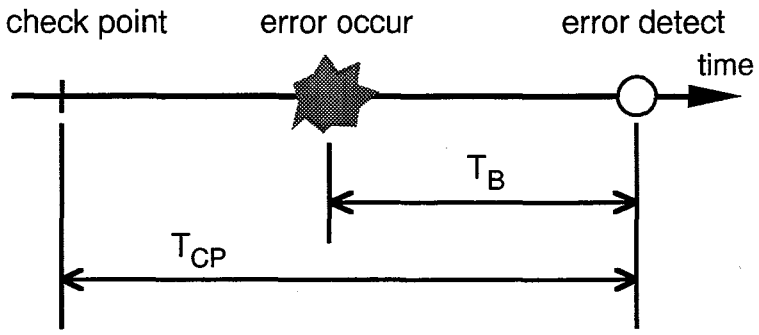
（１）障害レベルＡからのデータベースの復元を行う。障害レベルＢは復元の対象としない。

（２）ＣＰＤＢは数分おきに取得し、ＣＰＤＢ取得が完了した時点で、それ以前

表３.４　データの重要度に基づく分類

重要度レベル	復元レベル	情報例	リカバリ条件		
			ＬＯＧの要否	ＬＯＧ取得時間	ＣＰＤＢの要否
1.5	すべての障害に対し障害直前に戻す	なし *1	要	数分以上	要
1	ほとんどのハード／ソフト障害に対し障害直前に戻す	位置情報	要	数分以下	要
0.5	ＣＰ値に戻す	（位置情報）	否	—	要
0	初期値に戻す	回線の使用状況表示フラグ	否	—	否

*1　バンキングシステムの”預金額”が相当する



T_B：エラー発生から障害検出までの時間

T_{CP}：チェックポイントデータベース取得から
障害検出までの時間

データベースのリカバリ条件

$T_{CP} > T_B$

図3.7　T_{CP}，T_Bの関係

に取得したＬＯＧ及び旧のＣＰＤＢは破棄する。

（３）障害発生時にはＣＰＤＢとＬＯＧを用いて障害直前のデータベースを約１分以内でメモリ上に復元する。

（４）ＣＰＤＢ及びＬＯＧの格納媒体については、二重化された半導体ディスク装置を用い、バッテリによりデータの不揮発化を図る。

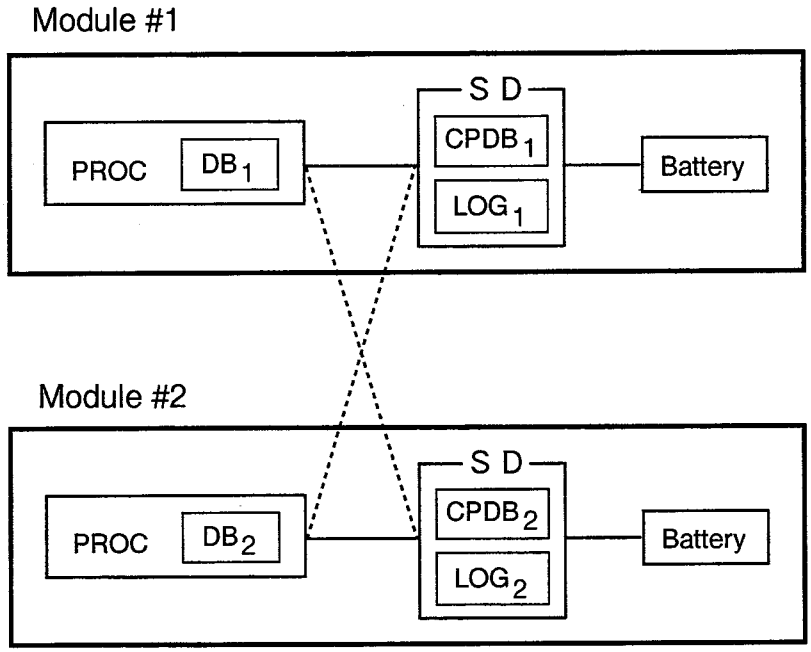
データベースのリカバリのためのハード構成を図３.８に示す。データベースのバックアップ方式には、通信によりＬＯＧをバックアップする別モジュールへ送り込む方式がある[18]。ここで提案した入出力による方式は、通信による方式に比べて、以下の点で優れている。

（１）入出力処理のためのＤＳはＬＯＧ送受信のためのＤＳに比べて数分の１ですむ。

（２）入出力のデータ転送速度は、通信処理のデータ転送速度に比べて高速であるため、スループットが大きく、レスポンスが短い。

（３）不揮発化された半導体ディスク装置にデータベースが格納されており、ノード全体での電源断に対してもリカバリが可能である。

メモリデータベースのリカバリ方式に関し、入出力による方式と通信による方式の適用領域や実用化に向けた課題とその対応策については第６章で述べる。



PROC：Processor
SD：Semiconductor disk
DB：Current Database
CPDB：Database at checkpoint
LOG：Log data since checkpoint

図3.8　データベースのリカバリとモジュール間相互バックアップ

3.4.2　冗長化構成法

3.4.2.1　冗長化構成

冗長化の方法はモジュールの機能に応じて異なるため、以下のようにモジュールを２つのタイプに分類した。

【タイプⅠ】同一機能に属するモジュール群の中では、どのモジュールでも処理可能なモジュール。ＳＣＦを実現するモジュールや、参照系データベースに多重方式を適用したＳＬＰディレクトリ検索機能を実現するモジュールがこれに該当する。

【タイプⅡ】同一機能のモジュールであっても、特定のモジュールのみが処理可能なモジュール。更新系データベースに分割方式を適用したＳＤＦを実現するモジュールがこれに該当する。

以下、モジュールの各タイプ毎に、ノード内及びノード間での冗長化構成法について述べる。

（１）ノード内冗長化

タイプⅠについては、同一機能を有するモジュール全体が障害となったモジュールの処理を肩代わりする。すなわち、同一機能に属するモジュール全体として、プロセッサの使用率が許す範囲でグループとしてバックアップする。

タイプⅡについては、１対１のモジュールの組を構成し、相互にバックアップする。具体的には、障害となったモジュールの半導体ディスク装置を別のモジュールが引継ぎ、データベース処理を再開する。モジュールと半導体ディスク装置間のチャネル数が少なく、モジュール、半導体ディスク装置の構成が容易であること、及びソフト制御が簡単なことから１対１のバックアップ構成とした。モジュール間バ

ックアップの概略構成を図３．８に示す。

（２）ノード間冗長化

地震、火災等の重度な障害及びノード内の複数モジュール障害に対してもアベイラビリティの向上をはかるため、地理的に離れて設置された２つのノード間で相互にバックアップを行うこととした[23][24][43]。図３．９に２つのノード間でのバックアップ構成を示す。これら２つのノードは全く同じモジュール構成を取っている。

タイプＩについては、２つのノード間でも同一機能を有するモジュール全体で相互にバックアップする構成をとる。タイプⅡでは、ノード内ではモジュール相互に１対１のバックアップ構成をとるが、さらに、２つのノード間で対応するモジュール相互に１対１のバックアップ構成を取っている。２つのノード間の対応するモジュール間では正常時には、ＤＢのＬＯＧを相互に送出し、バックアップのためのＤＢを更新している。

３．４．２．２　制御方式

全てのモジュールについて、正常時には、プロセッサ能力の半分を使用し、残りの半分はバックアップ用とした。タイプＩの場合、同一機能のモジュール群の中で半分以上のモジュールが障害となり、要求されるトラヒックを処理できなくなった時点で、障害となったノードのサービスはバックアップする側のノードに全て移される。タイプⅡの場合は、ノード内で１台のモジュールが障害となった時は、ノード内のバックアップするモジュールが障害となったモジュールのトラヒックを引き継ぐ。さらに、バックアップしているモジュールも障害となった時点で、障害となったノードのサービスはバックアップする側のノードに全て移される。

地震、火災等の重度な障害でノード全体が一度にサービスができなくなった場合は、その時点で、障害となったノードのサービスはバックアップする側のノードに

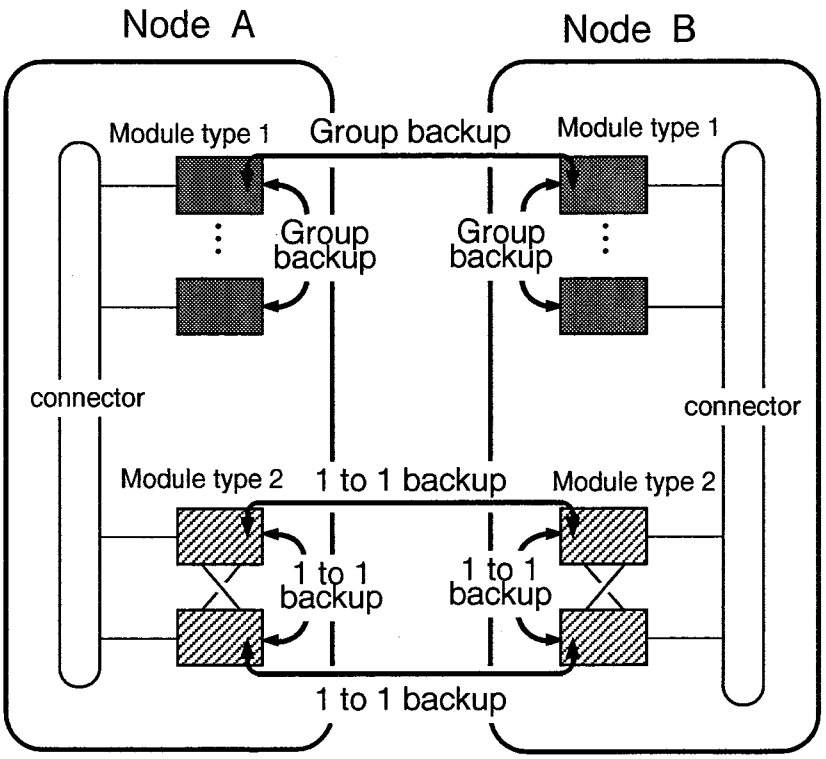


図3.9　ノード間バックアップ

全て移される。

モジュール相互のバックアップとノード相互のバックアップが同時に発生する確率は極めて低いことから、ノード内と２つのノード相互のバックアップのためのプロセッサ能力の余裕を共用させ、経済的な冗長構成を可能とした。ノード内のモジュール相互のバックアップと２つのノード相互のバックアップを行うことにより、１分／２０年以下の不稼働率を達成できる見通しを得ている。

３.５　分散処理ノード構成法

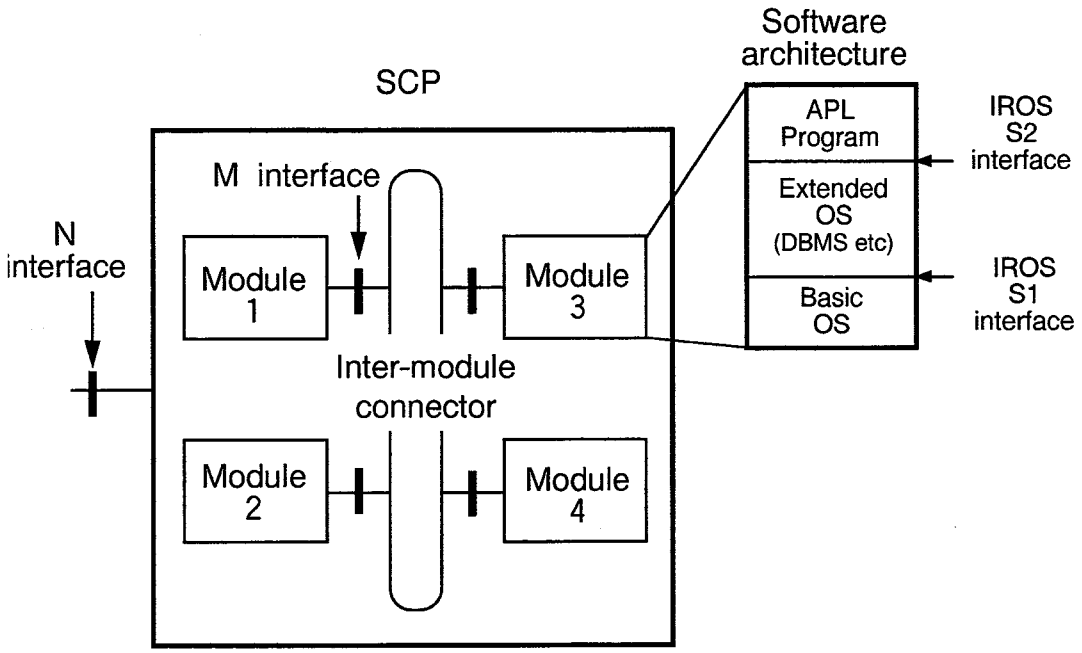
前節までに述べた方式を組み込んで試作した高度ＩＮ用ＳＣＰのアーキテクチャ及びハードウェア構成について述べる。

３.５.１　アーキテクチャ

交換機との通信はＴＣＡＰ（Transaction Capabilities Application Part）、ＳＭＳとの通信はＴＭＮ（Telecommunication Management Network）で行う。ノード内はＡＴＭレイヤをベースとし、ノード間通信と同一にするため、呼処理系はＴＣＡＰ、保守運用系はＴＭＮを使用したモジュール間インタフェースを規定し、各モジュール間はこのインタフェースを介して通信を行う。また、各モジュール内のソフトウェアについては、階層化を進め、基本ＯＳ（Ｓ１）、拡張ＯＳ（Ｓ２）インタフェースにそれぞれＩＲＯＳ（Interface for Realtime Operating Systems）[44][45]を採用した。ＳＣＰのアーキテクチャを図３.１０に示す。

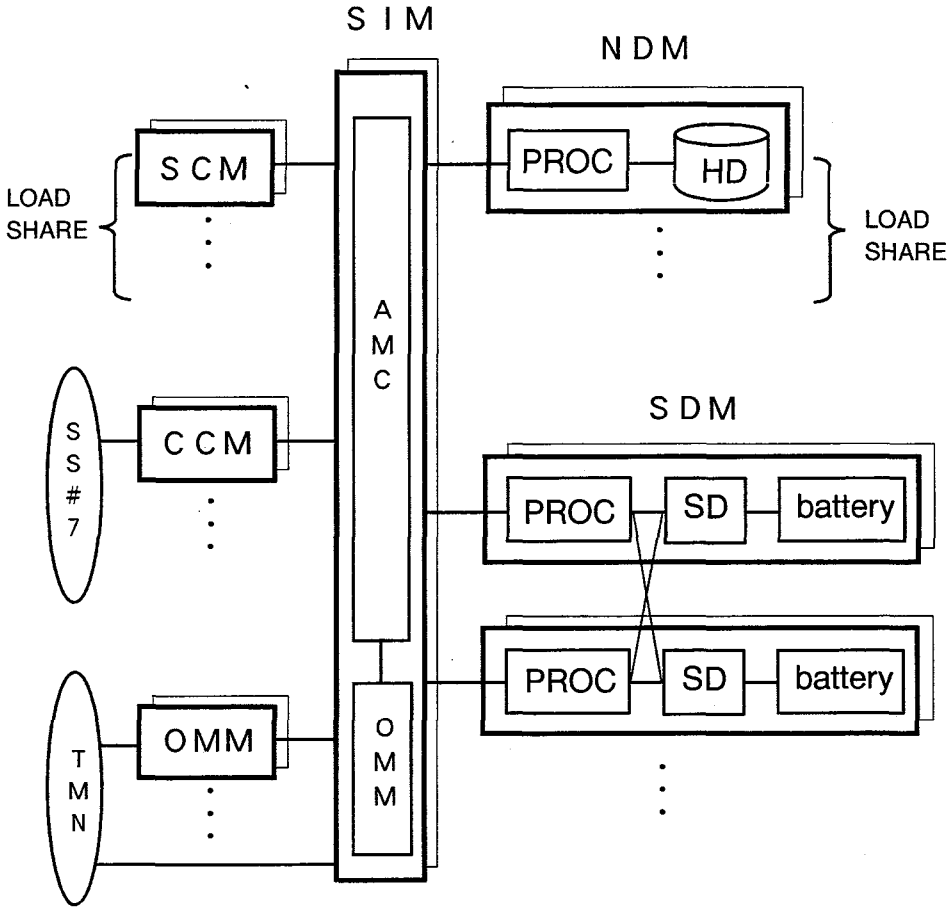
３.５.２　ハードウェア構成

ＳＣＰのハードウェア構成を図３.１１に示す。ＳＣＦはサービス制御モジュール



N interface：Inter-Node Interface
M interface：Inter-Module Interface

図３．１０　ＳＣＰのアーキテクチャ



NDM：network directory module
SDM：service data module
SCM：service control module
CCM：communication control module
OMM：operation and maintenance module
SIM：system interface module
AMC：ATM connector

図3.11　SCPのハードウェア構成

（ＳＣＭ）、ＳＤＦはサービスデータモジュール（ＳＤＭ）、ＳＬＰディレクトリ検索機能はネットワークディレクトリモジュール（ＮＤＭ）で実現した。通信処理には共通線信号網を介した交換機との通信とパケット網を介したＳＭＳとの通信があるが、各網の将来の発展形態が異なることから、それぞれ通信制御モジュール（ＣＣＭ）、オペレーション管理モジュール（ＯＭＭ）で分担した。モジュール間の結合機構としては、大規模ノードに十分なスループットを確保することからＡＴＭ結合機構（ＡＭＣ）を採用した。ＡＴＭスイッチには、１入力当たり１５６Ｍｂｐｓのデータ転送能力を有する１６×１６のスイッチを用いた。ＡＭＣ全体では２．４Ｇｂｐｓのデータ転送速度を持つ。運転・保守機能を実行するためのプロセッシング能力は小さくて済むため、ＯＭＭと共用化し、ＡＭＣとＯＭＭを一体化してシステムインタフェースモジュール（ＳＩＭ）で実現した。

３.６　結言

サービス制御ノードのデータベースへのアクセス特性、モジュールの負荷やその偏り等を総合的に考慮した、データベースの分散構成方式の評価方法を明らかにした。具体的には、複数のモジュールからのデータベースの共用の可否、主メモリ／半導体ディスク装置いずれかの格納媒体の選択に注目して構成案を設定した。これら代替案に対して、モジュール間の負荷の偏り、データベース処理でのダイナミックステップ数に対する評価を加え、コスト、性能面から総合的に評価する方法を明らかにした。これに基づき、サービスデータについては分割方式、ＳＬＰディレクトリについては多重方式による分散型メモリデータベース構成を提案した。データベースの高信頼化に関しては、ソフトウェアのバグやハードウェア障害によりデータベースが破壊された場合、データベースの復元の厳密さと復元時間に着目してデータ毎の重要度に基づいたリカバリレベルを設定し、サービス制御ノードのメモリデータベースに適した効率的なリカバリ方法を提案した。また、ノード内のモジュ

ール間バックアップとノード間バックアップを効率的に組み合わせ、高いアベイラビリティを経済的に実現する方法を明らかにした。さらに、モジュール間の結合機構としてＡＴＭを用い、これらの技術を総合して試作したサービス制御ノードの全体構成を示した。

第４章
モジュール間の負荷の偏りの評価技術

４.１　緒言

本章では、分散構成における方式設計や設備設計を効率的に行うため、データベースを多数のモジュールに分散配備した場合のモジュール間の負荷の偏りの評価方法を提案する[46]～[48]。

フリーダイヤル等の全国規模の高度ＩＮサービスを展開するためには、サービスデータを複数のモジュールに分散配置し、大量のトランザクションを分散処理することが必要となる。モジュール間の負荷の偏りの評価方法を明らかにすることは、少数のサービスデータへトラヒックが集中する傾向にあるサービス制御ノードのデータベースの分散構成を経済的に実現する上で有用である。特に、新しくシステムを設計する段階では、通常、サービス内容も大幅に更新されており、ユーザ特性が明確でない場合が多い。このような中でシステムの基本的な構造となる分散構成を決定してゆくためには、モジュール間の負荷の偏りについて大まかにでも事前に予測できる方法が必要となる。

本章では、カスタマ毎のサービスデータに対するトラヒック比が数千倍から数万倍と大きくばらつく場合のトラヒック量を大きいものから順位付けし、その順位によりカスタマ毎のトラヒック量を近似する方法を提案する。これに基づき、複数のモジュールにサービスデータを分散配備した場合のモジュール間の負荷の偏りの評価方法や、上限値の推定方法を明らかにする。また、現状のプロセッサ性能、高度ＩＮサービスにおけるトランザクションの処理量などから負荷の偏りが１～２割程度になることを明らかにする。さらに、この程度の偏りに対しては、ＳＬＰディレクトリ検索処理とサービスデータに基づくデータベース処理を組み合わせることにより、負荷を動的に平準化することが有効であることを示す。

4.2　対象とする分散処理モデル

フリーダイヤルサービス、パーソナル通信サービスなどの全国的な高度ＩＮサービスを対象とする。システム構成としてはネットワーク内のＮカ所にノードを広域的に配置し、１つのノードは更にM台のモジュールで構成する。分散システム構成を図４.１に示す。

4.2.1　処理方式

高度ＩＮサービスを受ける呼（以下、ＩＮ呼と記す）はユーザからの発信（フリーダイヤルサービスの場合、「０１２０＊＊＊＊＊＊」のダイヤル）により生起される。発信されたＩＮ呼は交換機（ＳＳＰ）で受信され、共通線信号網を介してトランザクションがサービス制御ノード（ＳＣＰ）に送られる。ＳＣＰではサービスデータを用いてトランザクション処理を行い、その結果をＳＳＰに送信する。ＳＳＰではそれに基づき呼の接続処理、切断処理等を行う。ＳＣＰへのトランザクションとしては、ＩＮ呼を接続、切断するため交換機から送られるトランザクションの他にサービス条件等を変更するためサービス管理ノード（ＳＭＳ）から送られるトランザクションがある。交換機からのトランザクションに比べてＳＭＳからのトランザクションはトラヒック量が２～３桁少なく、ＳＣＰへのトラヒックとしては交換機からのトランザクションのみを考えれば十分である。本章ではＩＮ呼の接続処理、切断処理のトランザクションのみを対象とする。

サービスデータを複数のモジュールに分散して配置する場合、トランザクション処理の前にサービスデータが配置されているモジュールを特定する必要がある。本章では、サービスデータの配置先モジュールを示すディレクトリをすべてのＳＣＰに配置する。ＳＳＰは前もってＳＳＰ毎に定められたＳＣＰにアクセスする。そのＳＣＰでディレクトリ検索処理を行い、サービスデータが配置されているＳＣＰとＳＣＰ内のモジュールを特定し、そこへトランザクションを転送する。ディレクトリ検索処理

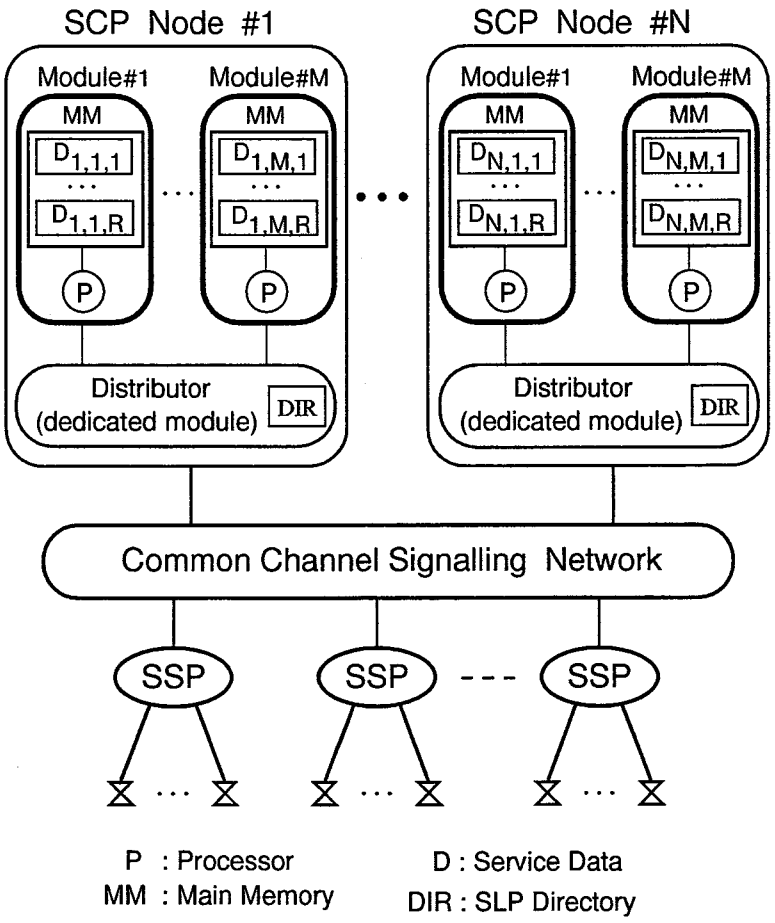


図4.1　分散システム構成

はＩＮ呼当たり１回行う。ＩＮ呼を処理するため複数のトランザクション処理が必要な場合、ディレクトリ検索処理は最初のトランザクションのみ行い、２回目以降のトランザクションは直接サービスデータが配置されているモジュールに送る[5]。

１つのＩＮ呼を処理するためのトランザクション数はサービス内容にも依存するが、フリーダイヤルサービスの場合は２～３回程度であり、フリーダイヤルサービスのＩＮ呼毎のばらつきは小さい。

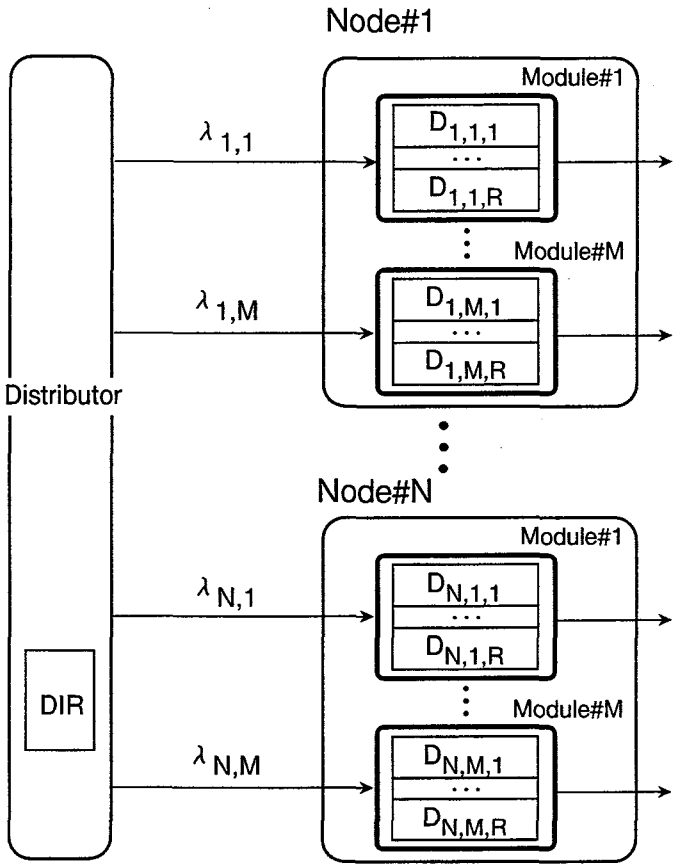
モジュールの負荷はトランザクション当たりの処理量とモジュールに配置したサービスデータへのトラヒック量の総和との積となる。１つのトランザクションの処理はサービスデータへの参照、更新などの処理がほとんどであり、トランザクション毎の処理量の差は小さい。本章ではすべてのトランザクションの処理量は一定とする

フリーダイヤルサービスを例にとると、１つの回線に接続するＩＮ呼の数がカスタマ毎に大きく異なる。さらに、ユーザからのＩＮ呼を受信する回線は１回線のところが大半であるが、大量のＩＮ呼を同時に受け付け可能とするため、カスタマによっては数１０から数１００の回線を設置する場合がある。これらの回線へＩＮ呼を接続するためのトランザクションはモジュールに配置されたサービスデータを用いて処理され、そのデータへのトラヒック量は非常に高くなる。このようにフリーダイヤルサービスなどの企業を対象としたサービスでは、極く少数のサービスデータへトラヒックが集中する傾向がある。

４.２.２　評価モデル

サービスデータへのトラヒック量のばらつきに起因するモジュール間の負荷の偏りの評価モデルを図４.２に示す。

各モジュールには一律にＲ個のサービスデータを無作為に抽出して配置する。サービスデータはどれか１つのモジュールにのみ配置される。サービスデータへのトラヒック量の予測などに基づき、モジュールに配置するサービスデータ数を調整する方法も考えられるが、サービスデータ配置操作の自動化が困難となること、季節的要因や



N : Number of nodes
M : Number of modules
R : Number of service data allocated in each module
DIR : SLP Directory
$D_{i,j,k}$: Service Data
$y_{i,j,k}$: Amount of traffic for the service data
$\lambda_{i,j}$: Amount of traffic for the module

$$\lambda_{i,j} = \sum_{k=1}^{R} y_{i,j,k}$$

(i : Node number , j : Module number , k : Service data number)

図４．２　評価モデル

経済的要因に基づく変動を予測することは難しいことから、無作為に抽出して配置するとした。

発生したトランザクションは、それを処理するために必要なサービスデータを配置しているモジュールに振り分ける必要がある。評価モデルでは振り分けのためのディレクトリ検索処理は別の専用モジュールで処理されるとした。評価モデルの各モジュールではサービスデータと関係するトランザクション処理のみを実行する。

4.3　サービスデータへのトラヒック量

4.3.1　サービスデータへのトラヒック量の近似

フリーダイヤルサービスにおけるカスタマの1カ月間のＩＮ呼数（ＩＮ呼数は度数に比例するとした）の測定結果をもとに、その分布例を図4.3に示す。図4.3の横軸はＩＮ呼数の高いカスタマからの順位を示し、縦軸はＩＮ呼数を示す。図中の黒い点はＩＮ呼数の高い順にカスタマを4つのグループに分割したときの各グループ内でのカスタマの順位、ＩＮ呼数の平均を示す。点線は各グループ毎のカスタマの占める範囲を示す。図4.3中の直線により、x番目のカスタマのＩＮ呼数Kは、おおよそ

$$K = Cx^{-\alpha} \quad (1 \leqq x \leqq h) \tag{4.1}$$

の関係で近似できる。ここで、hはカスタマの数である。αは図4.3中で直線の傾きを示し、カスタマ間のＩＮ呼数の偏りを示す係数と見なすことができる。以後、αを偏り係数と記す。Cはサービス特性等に依存して決まる比例定数である。サービスデータへのトラヒック量を式（4.1）を用いて近似した場合、図4.3中のフリーダイヤルサービスの4つの点の近似としては、おおよそ$\alpha = 0.8$と見なすことができる。また、$\alpha = 0.8$の場合、$x = 1$と$x = 10^5$のKの比は約10^4であり、ＩＮ呼数の高いカスタマと低いカスタマとのＩＮ呼数の比は約10^4となる。

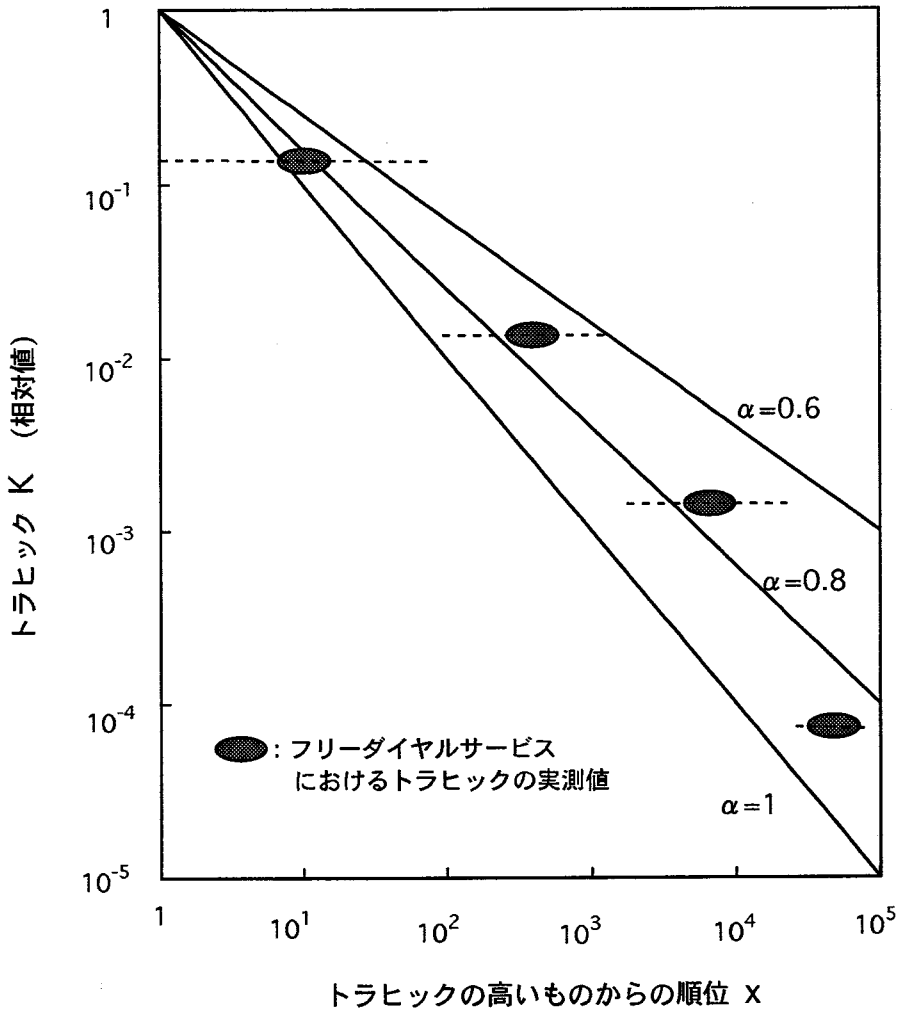


図4.3　カスタマ毎のＩＮ呼数

時々刻々変化するサービスデータへのトラヒック量を直接測定することは容易でないため、以下のように仮定する。

（1）ＩＮ呼当たりのトランザクション数は一定とする。

（2）一般にフリーダイヤルサービスなどのＩＮ呼の発生はある時間帯（たとえば、会社が営業を行っている時間など）に集中する傾向がある。どのカスタマについても、ＩＮ呼が集中する時間帯および集中するＩＮ呼の割合は同一とし、かつこの時間帯内ではＩＮ呼は一様に分布するものとする。

ノードやモジュールの設備容量の設計ではトラヒックが集中する時間帯を重点的に考慮するため、上記仮定に基づいて、サービスデータへのトラヒック量ｙをトラヒック量の高いものからの順位ｘを用いて

$$y = cx^{-\alpha} \qquad (1 \leqq x \leqq h) \tag{4.2}$$

の関係で近似する。ここで、最もトラヒック量の高いサービスデータ（ｘ＝１）と最もトラヒック量の低いサービスデータ（ｘ＝ｈ）とのトラヒック量の比（以下、最大トラヒック量比と記す）をｇとおくと、$h = g^{\frac{1}{\alpha}}$となる。

語彙論において、ある文学作品で使用されている言葉の使用度数ｙとその使用度数の大きい方からの順位ｘの関係を表すものとして式（4.2）と同様の関係式が使用されている。これはジップの法則として知られている[49]。

4.3.2　サービスデータの母集団の設定

サービスデータの取りうるｘの値は１からｈまでの整数値である。これを０から１までに正規化するため、

$$z = \frac{x-1}{h-1} \qquad (0 \leqq z \leqq 1)\ (1 \leqq x \leqq h) \tag{4.3}$$

とおき、式（4.2）に式（4.3）、$h=g^{\frac{1}{\alpha}}$を代入すると次式を得る。

$$y=c\left\{(g^{\frac{1}{\alpha}}-1)z+1\right\}^{-\alpha} \qquad (0 \leqq z \leqq 1) \tag{4.4}$$

ここまでの展開では、zは等間隔に分布するh個の離散値をとる。しかし、一般には通信サービスを受けるカスタマ数は非常に多く、z＝0からz＝1までの間に無数のサービスデータが稠密に分布していると考えられる。式（4.4）は順位を0から1までに正規化したサービスデータの母集団の特性を示すものとみなすことができる。以下、式（4.4）で表されるサービスデータの母集団を$U(\alpha,g)$で表す。式（4.4）はパレート分布と言われているものと同一の形態をとる。パレート分布は所得とそれを得る人数の関係を近似する方法として Pareto によって最初に提案されたものである[50]。パレート分布は高額所得者の分布とも言われ、高い所得を有する人数は少数であるが、全体の所得に占める割合が高い不均等の分布を良く近似できると言われている。

4.3.3 近似式の特性

母集団$U(\alpha,g)$から任意に1つのサービスデータを抽出した場合のトラヒック量の期待値y_0は次式で表せる（付録4.1参照）。

$$y_0=\int_0^1 y\,dz=\int_0^1 c\left\{(g^{\frac{1}{\alpha}}-1)z+1\right\}^{-\alpha} dz$$

$$=\begin{cases}\dfrac{c(g^{\frac{1}{\alpha}-1}-1)}{(1-\alpha)(g^{\frac{1}{\alpha}}-1)} & (\alpha \neq 1)\\[2ex] \dfrac{c\log g}{(g-1)} & (\alpha=1)\end{cases} \tag{4.5}$$

式（4.5）の$\alpha \neq 1$の式において、αを1に近づけた場合の極限値は$\alpha=1$の式と同一となる。このため、以後、αの値に係わらず式（4.5）の$\alpha \neq 1$の式を用いることとする。

次に、サービスデータへのトラヒック量の大きい方からｚまでの累積値が期待値に占める割合ηは次式で表せる（付録４．２参照）。

$$\eta = \frac{\int_0^z y dz}{y_0}$$

$$= \frac{(z - zg^{-\frac{1}{\alpha}} + g^{-\frac{1}{\alpha}})^{1-\alpha} - g^{1-\frac{1}{\alpha}}}{(1 - g^{1-\frac{1}{\alpha}})} \qquad (4.6)$$

ｚとｙとの関係を図４．４、ｚとηとの関係を図４．５に示す。図４．４より、$\alpha = 1$の場合、ｚ＝０からｚ＝０．１の間で、トラヒック量は１から10^{-3}まで急激に低下する。一方、ｚ＝０．１からｚ＝１の間では、トラヒック量は10^{-3}から10^{-4}まで緩やかに低下する。

αが１より大きくなるに従って、トラヒック量の高い領域での低下傾向は緩やかとなり、ｚの全領域に渡ってトラヒック量の低下の度合いが一定となってくる。これは、αが無限大となった場合、ｙが次式に収束することから裏付けられる（付録４．３参照）。

$$\lim_{\alpha \to \infty} y = ce^{-(l_{og}g)z}$$

$$= cg^{-z} \qquad (4.7)$$

一方、αが１より小さくなるに従って、トラヒック量の高い領域では一段と急激に低下する。但し、全体としてトラヒック量の高い領域が減少し、ほとんどがトラヒック量の小さいサービスデータのみの一様な分布に近づく。これは、αが０となった場合、次式に示すようにηがｚに比例して増加することから裏付けられる。

$$\lim_{\alpha \to 0} \eta = z \qquad (4.8)$$

以上のことから、式（4.4）はαを変化させることにより、一様な分布から指数関数的に低下の度合いが一定となる分布まで幅広く表現できることがわかる。

全体として、αが極めて小さい領域を除けば少数のサービスデータが大部分のトラヒック量を占める。たとえばｇ＝10^4の場合、図４．５より$\alpha = 1$、$\alpha = \infty$ともにトラヒック量の高いサービスデータの上位約２割で総トラヒック量の約８割を占める。モ

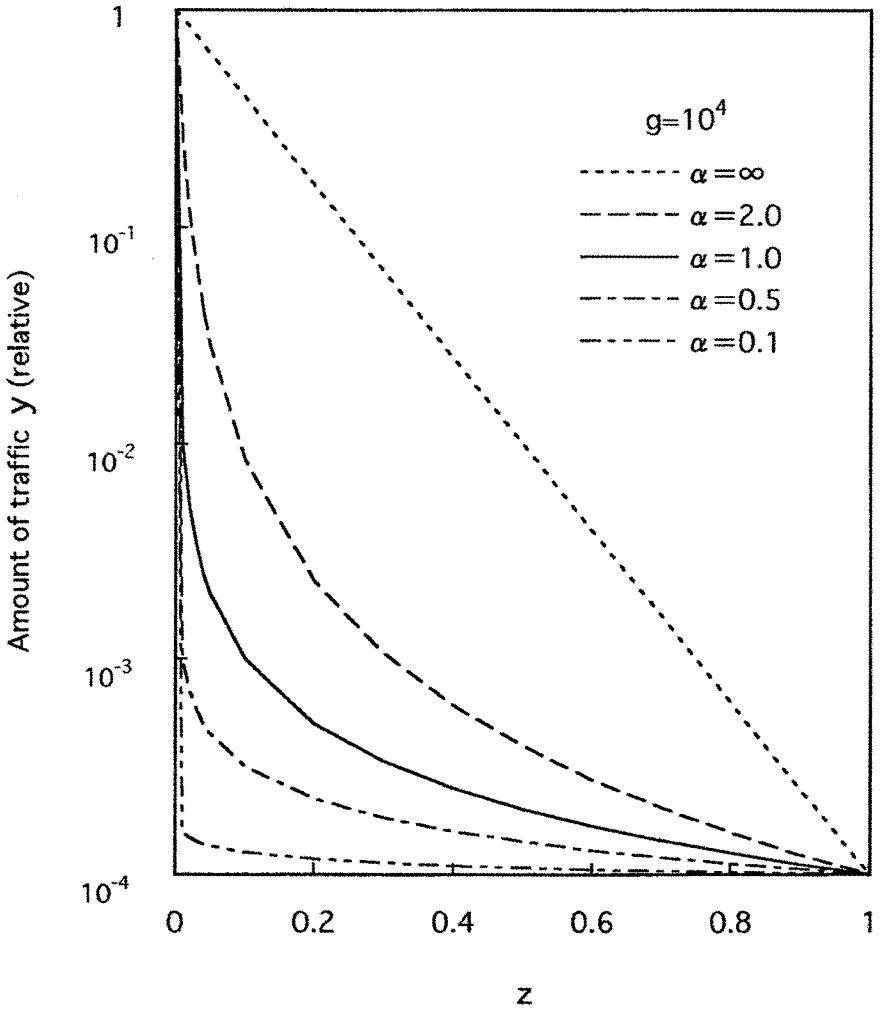

1
10^{-1}
10^{-2}
10^{-3}
10^{-4}
Amount of traffic y (relative)
0
0.2
0.4
0.6
0.8
1
z
$g=10^4$
$\alpha=\infty$
$\alpha=2.0$
$\alpha=1.0$
$\alpha=0.5$
$\alpha=0.1$

図4.4　トラヒック量

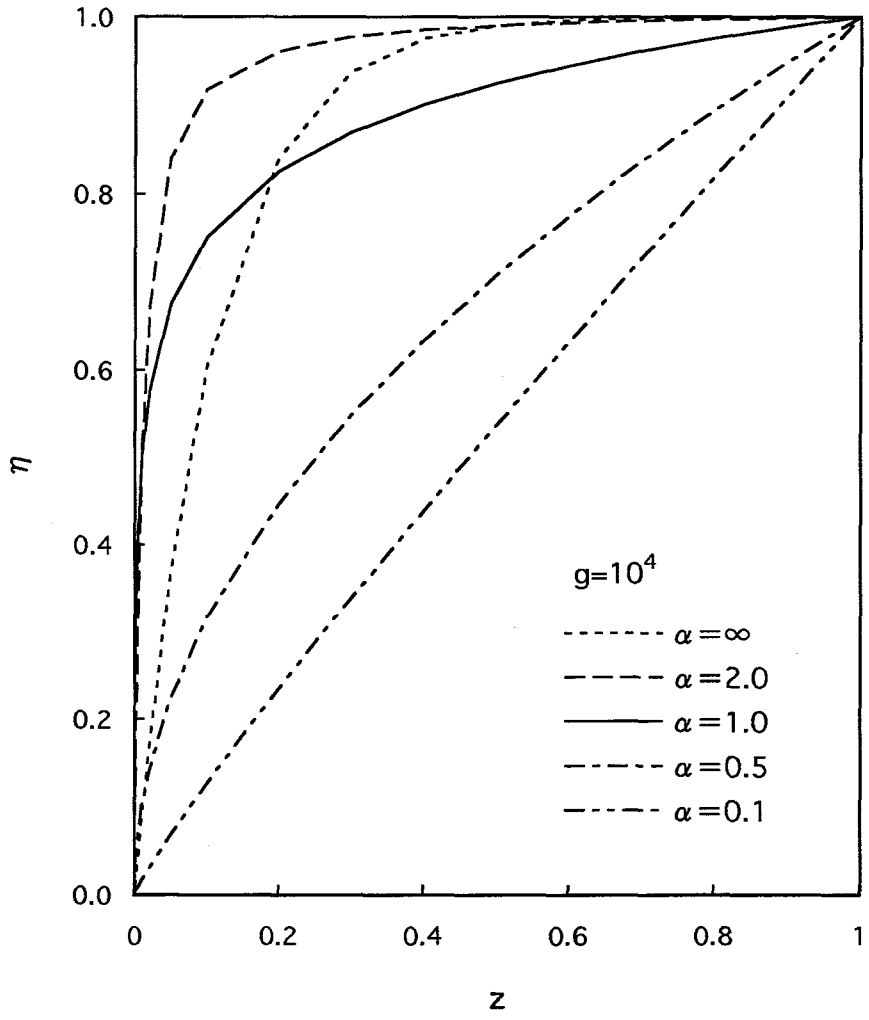


図４.５　ｚとηの関係

ジュール間の負荷の偏りに影響を及ぼすのはトラヒック量の高い少数のカスタマのばらつきである。図4.5ではzの小さい領域での分布が明確でないため、横軸を対数として図4.6に示す。図4.6よりz＝0.01の場合、$\alpha=1$では$\eta=0.5$であるが、$\alpha=\infty$では$\eta=0.1$となる。$\alpha=1$の分布は$\alpha=\infty$に比べて、極く少数のサービスデータへトラヒックが集中する分布を適切に近似できる。以上の考察に基づき、式（4.4）により、サービスデータへのトラヒック量を近似し、これを用いてモジュール間の負荷の偏りについて評価する。

4.4　モジュール間の負荷の偏り

4.4.1　負荷偏り率

$U(\alpha,g)$の中から無作為にサービスデータをR個抽出し、モジュールに配置する。R個のサービスデータへのトラヒック量をそれぞれ$cx_1^{-\alpha},cx_2^{-\alpha},\cdots,cx_R^{-\alpha}$とおくと、総トラヒック量$\lambda$は、

$$\lambda = cx_1^{-\alpha} + cx_2^{-\alpha} + \cdots + cx_R^{-\alpha} \tag{4.9}$$

となる。λの期待値をλ_0とおき、その偏りを$\frac{\lambda}{\lambda_0}$で表す。このとき標準偏差$\sigma$は次式で表せる（付録4.4参照）。

$$\sigma = \left\{ \frac{1}{(h-1)^R} \int_1^h \cdots \int_1^h \left(\frac{\lambda}{\lambda_0} - 1\right)^2 dx_1 \cdots dx_R \right\}^{1/2}$$

$$= \frac{A(\alpha,g)}{\sqrt{R}} \tag{4.10}$$

但し、$$A(\alpha,g) = \left\{ \frac{(1-\alpha)^2 (g^{\frac{1}{\alpha}} - 1)(g^{\frac{1}{\alpha}-2} - 1)}{(1-2\alpha)(g^{\frac{1}{\alpha}-1} - 1)^2} - 1 \right\}^{1/2} \tag{4.11}$$

モジュールの最大負荷を（1＋ω）λ_0で表し、ωを3σで近似する。λの分布が

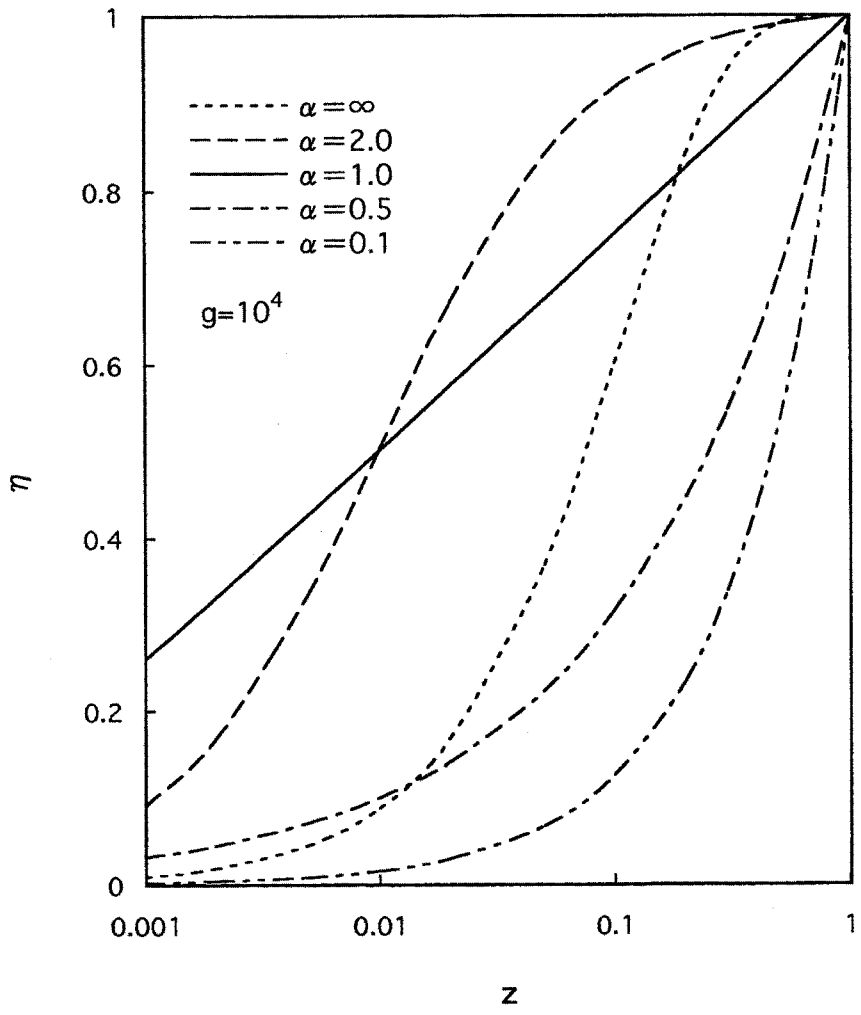


図4.6　zとηの関係

正規分布で近似できるとした場合、約９９.９５％のモジュールがこの最大負荷以内に収まる。数百モジュールが設置されたとしても、ほとんどのモジュールの負荷がこの最大負荷をこえることはない。以下、ωを負荷偏り率と記す。

$$\omega \approx 3\sigma = \frac{3A(\alpha, g)}{\sqrt{R}} \tag{4.12}$$

ωは$\sqrt{R}$に反比例し、その係数Aはα、gに依存して決まる。

4.4.2　サービスデータ数と負荷偏り率の関係

モジュールに格納するサービスデータ数Rと負荷偏り率ωとの関係を図4.7に示す。なお、図4.7では最大トラヒック量比gは10^4としている。図4.7より、α＝0.8の場合、Rとgが同程度、すなわちRが1万の場合は$\omega \fallingdotseq 0.3$と大きい。Rがgの１０倍になると$\omega \fallingdotseq 0.1$程度となり、さらに、Rがgの１００倍になると$\omega$は数%以下となり、無視できる程度となる。

gとωの関係を図4.8に示す。gが小さくなるに従ってωは小さくなる。たとえば、Rが１０万の場合、g＝10^4では$\omega \fallingdotseq 0.1$であるが、g＝$10^2$になると$\omega \fallingdotseq 0.02$に低下する。

図4.3のフリーダイヤルサービスの場合にはg＝10^4であったが、パーソナル通信サービスのように個人レベルの通話の場合にはgは小さくなり、それに伴ってωは低下する。たとえば、gがフリーダイヤルサービスに比べて１／１００程度になったとした場合、ωは数%以下となり、無視できる程度となる。

4.4.3　偏り係数と負荷偏り率の関係

負荷偏り率ωは、式（4.12）に示す通り、$\sqrt{R}$に反比例し、係数Aに比例する。サービスデータへのトラヒックの偏り係数αと負荷偏り率ωの係数Aとの関係を図4.9に示す。図4.9からわかる通り、gの値に係わらずα＝1でAは極大値をとる。これはAをαで微分し、$\alpha \rightarrow 1$とすると

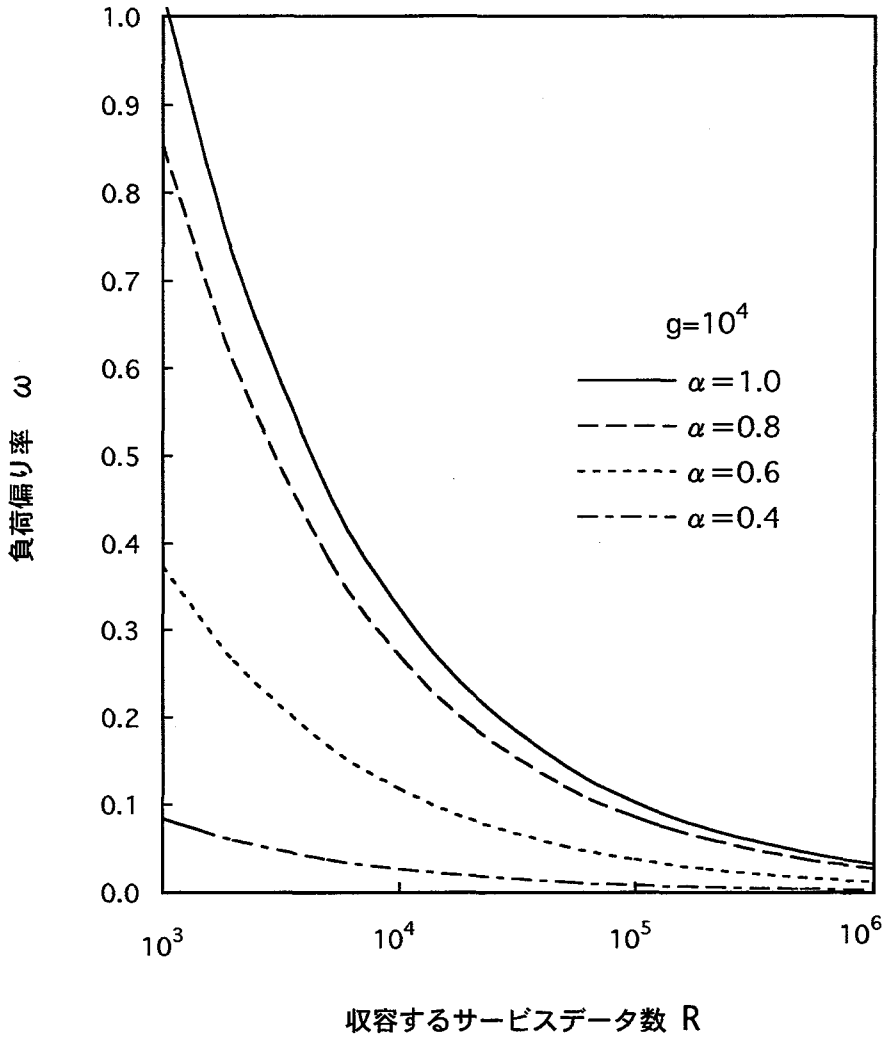

1.0
0.9
0.8
0.7
0.6
0.5
0.4
0.3
0.2
0.1
0.0
負荷偏り率 ω
g=10^4
α=1.0
α=0.8
α=0.6
α=0.4
10^3
10^4
10^5
10^6
収容するサービスデータ数 R

図4.7　Rとωの関係

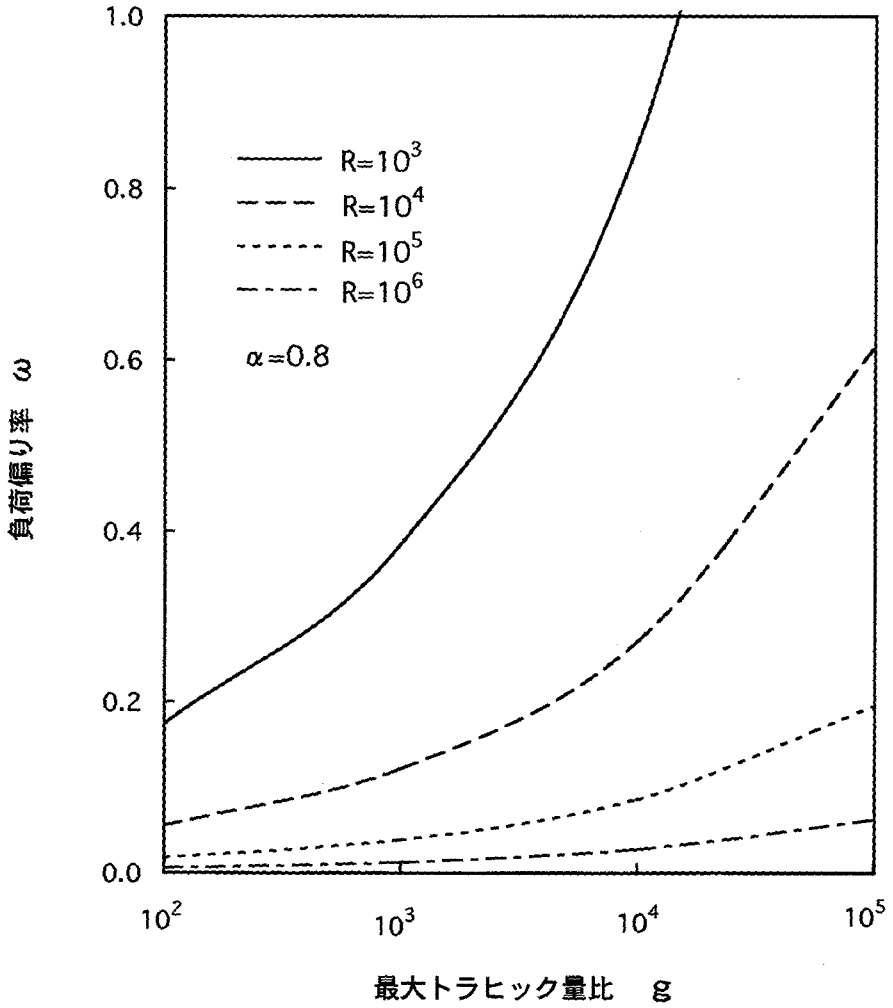

1.0
0.8
0.6
0.4
0.2
0.0
負荷偏り率 ω
R=10^3
R=10^4
R=10^5
R=10^6
α=0.8
10^2
10^3
10^4
10^5
最大トラヒック量比 g

図4.8　gとωの関係

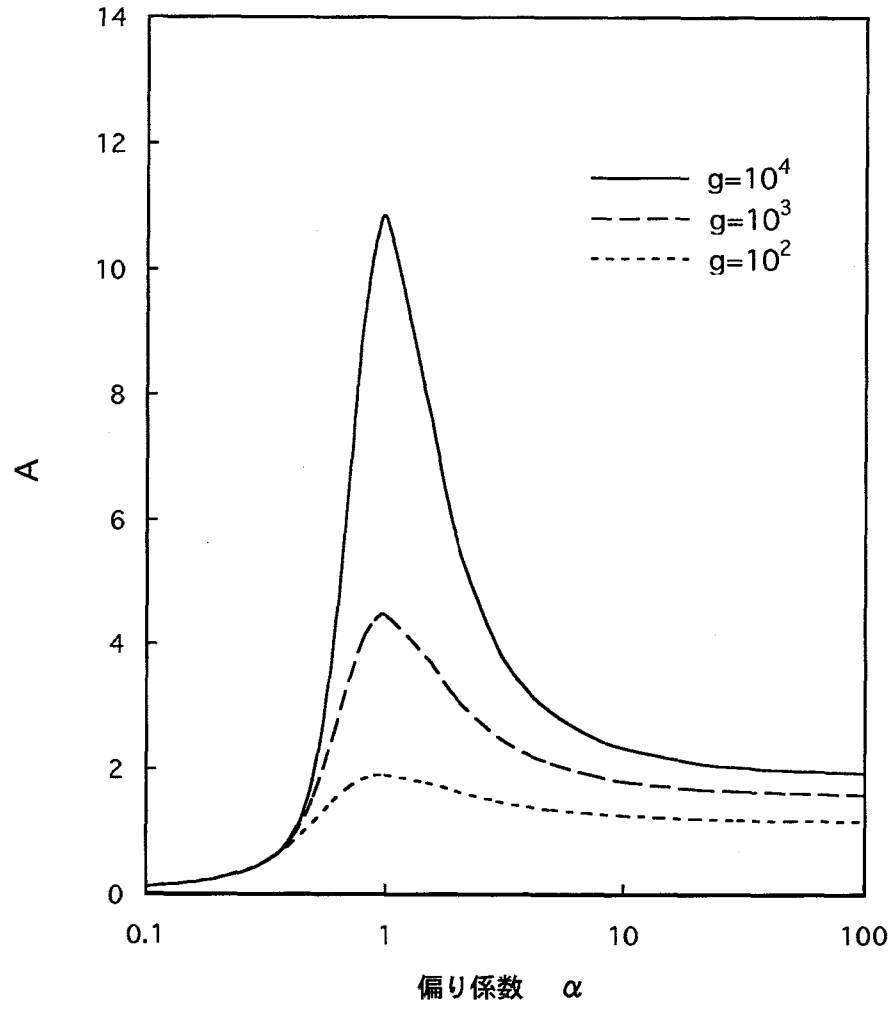


図４.９　αとＡの関係

$$\lim_{\alpha \to 1} \frac{dA(\alpha, g)}{d\alpha} = 0 \tag{4.13}$$

となる（付録４．５参照）ことから裏付けられる。αが０から１に近づくに従って、トラヒック量の小さな大多数のサービスデータ間でのトラヒック量の差が拡大するため、Ａは増加する。一方、αが１より大きくなると、高いトラヒック量をとるサービスデータ間のばらつきが縮小する傾向にあり、Ａは減少する。$\alpha = 1$と$\alpha = \infty$でのＡの比（$A(1,g)/A(\infty,g)$）は、ｇ＝10^4の場合、約６となる。

$\alpha = 1$でＡが極大値、すなわちωが極大値をとるため、αの値が明確でない場合でも、ｇについて推定できるときは$\alpha = 1$で評価することにより、ωの上限を把握することができる。なお、図４．３に示したフリーダイヤルサービスの場合、$\alpha = 1$と$\alpha = 0.8$のときのωの差は約２０％程度である。

４．５　分散構成への適用

４．５．１　負荷の偏りの影響

前節までの評価方法を用いて、現状のプロセッサ性能とトランザクション処理のダイナミックステップ数を基に分散構成への影響を明らかにする。

最も負荷の高いモジュールの負荷をL_{MAX}、プロセッサ使用率をρ_{MAX}、プロセッサ性能をＰとすると次式が成り立つ。但し、ここで負荷はモジュールへのトラヒック量とダイナミックステップ数（Ｄ）との積とした。

$$L_{MAX} = (1+\omega) R y_0 D \tag{4.14}$$

$$\rho_{MAX} = \frac{L_{MAX}}{P} \tag{4.15}$$

最も負荷の高いモジュールについても要求されるレスポンスタイムを保証するためにはＲを少なくしてプロセッサ使用率ρ_{MAX}を０．８程度以下に抑える必要がある。Ｒは式（4.12）、（4.14）、（4.15）より次式となる。

$$R = \rho_{MAX}\frac{P}{y_0 D} - \left(\sqrt{(3A)^2\rho_{MAX}\frac{P}{y_0 D} + \frac{(3A)^2}{4}} - \frac{(3A)^2}{2}\right) \tag{4.16}$$

式（4.16）の第１項は負荷の偏りが無い場合の配置可能なサービスデータ数（R_0）、第２項は負荷の偏りに対応するため減少すべきサービスデータ数を表している。

なお、サービスデータはサービス内容にも依存するが最大でも数１００バイト～１０００バイト程度と評価しており、サービスデータ数が数万～数十万程度までは、主メモリ容量からの制約はほとんど無い。

ＰとＲの関係を図４．１０に示す。図４．１０ではモジュール間の負荷の偏りが無い場合、すなわち$\alpha = 0$のときの収容可能なサービスデータ数R_0との相対値（R/R_0）で表している。現状のプロセッサ性能、ダイナミックステップ数およびサービスデータへのトラヒック特性から、Ｄ＝１００キロステップ、Ｐ＝２０ＭＩＰＳ、ｇ＝10^4、$y_0 = 3\times10^{-3}$トランザクション／秒／サービスデータとすると、モジュール間の負荷の偏りに関しては以下のことが言える。

（１）$\alpha = 0.8$の場合、Ｒ≒４．８万、$\omega \fallingdotseq 0.12$となる。サービスデータ間の最大トラヒック量比ｇが$10^4$と大きく、かつ各サービスデータへのトラヒック量の予測などを行わず無作為に抽出して同数配置したとしても、Ｒが５万程度になるとモジュール間の負荷の偏りを１～２割程度に抑えることができる。$\alpha = 1$の場合はモジュール間の負荷の偏りが極大となるが、Ｒ＝４．７万であり、$\alpha = 0.8$の場合と比べてほとんど差はない。

（２）ノード全体を１つの単位として見たとき、ノード間の負荷の偏りもモジュール間の負荷の偏りと同様に評価することができる。ノード間の負荷偏り率をω_nとすると、ノード内にはＭ・Ｒ個のサービスデータが配置されていることから、

$$\omega_n = \frac{3A(\alpha, g)}{\sqrt{M \cdot R}} \tag{4.17}$$

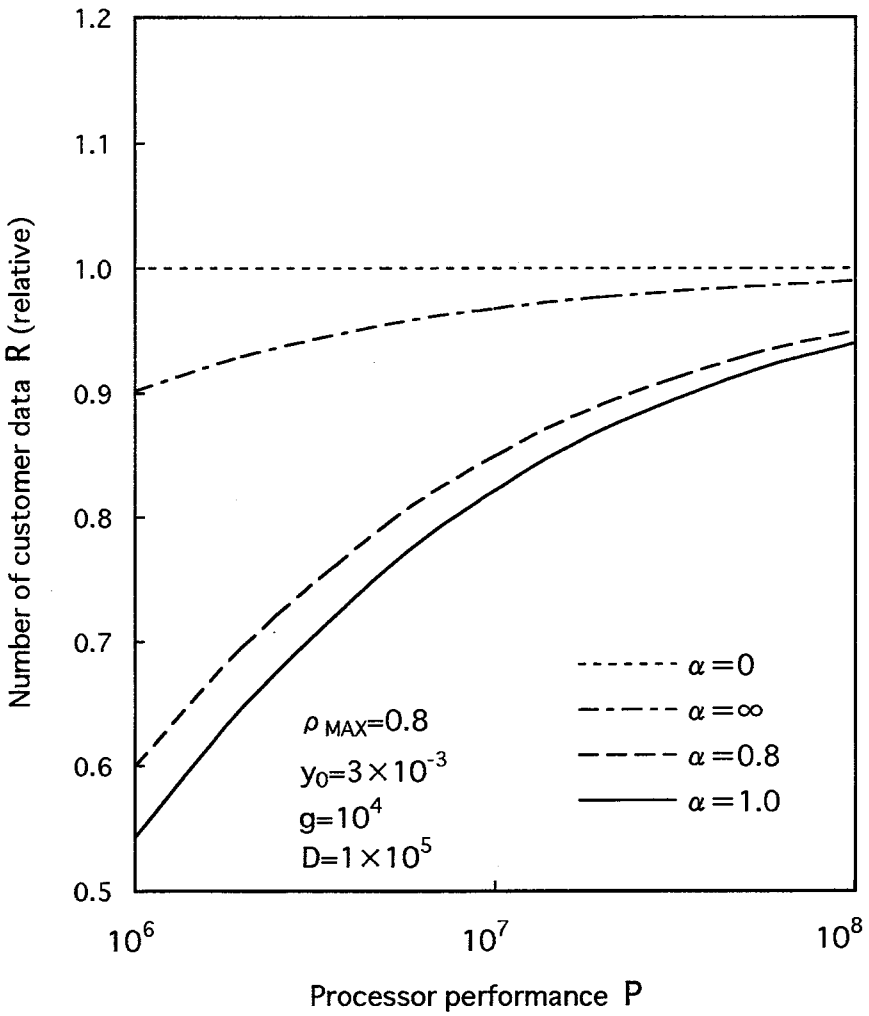

1.2
1.1
1.0
0.9
0.8
0.7
0.6
0.5
Number of customer data R (relative)
ρ_{MAX}=0.8
$y_0=3\times10^{-3}$
$g=10^4$
$D=1\times10^5$
α=0
α=∞
α=0.8
α=1.0
10^6
10^7
10^8
Processor performance P

図４．１０　ＰとＲの関係

と表せる。Rを5万程度、ノード内のモジュール数を20程度として、ノード内のサービスデータ数を100万（M・R＝10^6）とすると、ω_nは数%以下となり、ノード間の負荷の偏りは無視できる程度となる。

以上、フリーダイヤルサービスを中心にモジュールに格納できるRについて考察した。式（4.16）はサービスデータからなるデータベースを複数のモジュールに分散収容したシステムであればサービスに依存せず適用できる。R／R_0はP／y_0Dに依存し、Pが大きくなるに従って大きくなり、y_0Dが大きくなるに従って小さくなる。現状のプロセッサ性能を想定すると、ダイナミックステップ数の大きい、より高度なサービスや、平均的なトラヒック量の高いサービスの場合には、y_0Dが大きくなりR／R_0が減少する。また、将来プロセッサ性能が高くなった場合でもサービスの高機能化に伴い、ダイナミックステップ数の増加がプロセッサ性能の向上度より大きい場合にはR／R_0が減少する。

4.5.2　負荷平準化の可能性

前節まで無作為に抽出したサービスデータでの負荷の偏りを評価してきたが、次にモジュール間での負荷の平準化の可能性について考察する。

負荷の平準化を実施するとした場合、サービス開始前にカスタマ毎の事前の予測に基づいて負荷が平準化するように、サービスデータを配置するモジュールを選定する方法とサービス開始後、動的に負荷を平準化する方法とが考えられる。前者については、4.2節で述べたように、データ配置操作の自動化が困難となること、季節的要因や経済的要因に基づく変動を予測することが難しいことから実用的ではない。モジュール間の負荷を動的に平準化する方法としては以下の2案が考えられる。

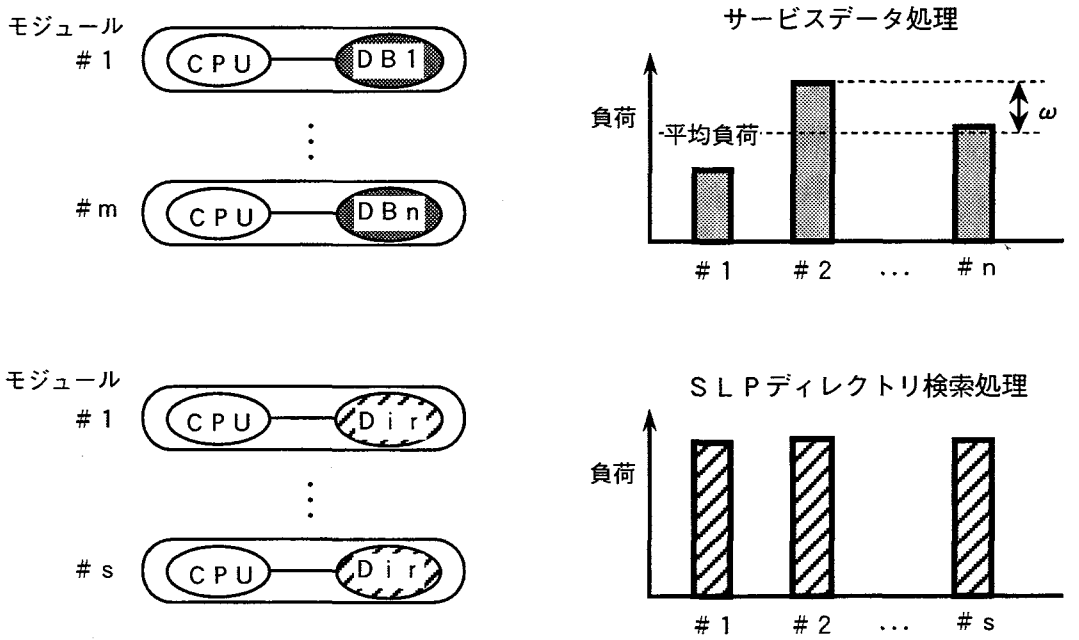
【方法1】ノード内のモジュール間で負荷の高いモジュールから負荷の低いモジュールへサービスデータそのものを移動し、モジュール間の負荷を平準化する。

【方法２】ノード内にサービスデータに結びつく処理だけでなく、サービスデータと独立に実行可能な処理を共存させ、後者をその時点で負荷の低いモジュールに多く割り当てることにより、モジュール間の負荷を平準化する。

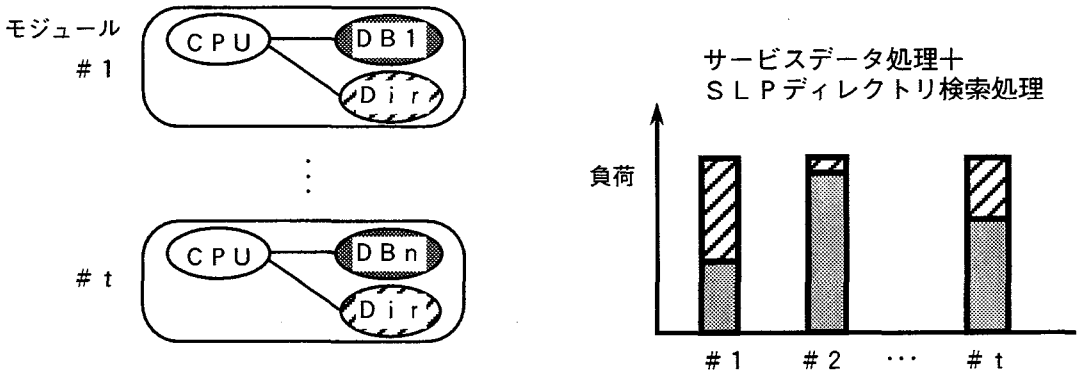
方法１は確実に負荷を平準化できるが、サービスデータの移し換えのためには、当該サービスデータの一時的な閉塞とモジュール間の転送が必要になるなど、プロセッサのオーバヘッドが大きく、頻繁な実行や高負荷時の実行には問題が多い。また、サービスデータの再配置により各ＳＣＰが持つＳＬＰディレクトリの変更処理も必要となる。このため、夜間、負荷が低いときなどに実施する必要があり、負荷の動的な変化には十分に対応できない。

方法２でどの程度までモジュール間の負荷を平準化できるかは、サービスデータと独立に実行可能な、どのモジュールにも割り当てできる処理量に依存する。負荷平準化のためのプロセッサオーバヘッドは小さく、負荷の動的な変化には十分対応できると考えている。

方法２において、サービスデータと独立に実行可能な処理として、４.２節で述べたＳＬＰディレクトリ検索処理がある。図４.２の評価モデルでは別の専用モジュールでＳＬＰディレクトリ検索処理を行い、トランザクションの振り分けを行うとして分離したが、この処理を各モジュールのトランザクション負荷に応じて動的に割り当てることを考える。具体的には、ＳＬＰディレクトリを各モジュールに重複して配置し、任意のモジュールがＳＬＰディレクトリ検索を実行可能とする。負荷の平準化方法を図４.１１に示す。図４.１１で機能別処理とは、サービスデータ処理とＳＬＰディレクトリ検索処理をそれぞれ別のモジュールで処理するとした方法である。ＳＬＰディレクトリ検索処理はどのモジュールでも可能であり、ラウンドロビン方法等によりモジュールの負荷を一様とすることができる。しかし、サービスデータ処理は最も負荷の高いモジュールに合わせてモジュールの稼働率を設定する必要があり、全体としてモジュールの稼働率が低下する。負荷の平準化を行うためには、１モジュールにサービスデータとＳＬＰディレクトリの両方を格納する必要があり、メモリやファイ



（１）機能別処理



（２）負荷平準化処理

<u>平準化方法</u>：サービスデータ処理とＳＬＰディレクトリ検索処理を組合わせ，サービスデータ処理の小さいモジュールにＳＬＰディレクトリ検索処理を多く割当てる．

DBi：サービスデータ
（更新系データベースで分割して各モジュールに配備）

Dir：ＳＬＰディレクトリ
（参照系データベースで全てのモジュールに重複して配備）

図４．１１　負荷の平準化方法

ル容量は多く必要となるが、モジュールの稼働率を高くでき、機能別処理に比べて少ないモジュール台数でサービスデータ処理とＳＬＰディレクトリ検索処理を行うことができる。負荷の平準化がどの程度行えるかはサービスデータ処理とＳＬＰディレクトリ検索処理の処理量や、サービスデータ処理でのモジュール間の負荷偏り率に依存する。ＳＬＰディレクトリ検索処理はＩＮ呼毎に必要であり、そのダイナミックステップ数はサービスデータに関係するトランザクション処理のダイナミックステップ数と同程度である。ＩＮ呼当たりのトランザクション数を２～３回とすると、ＳＬＰディレクトリ検索処理の処理量はサービスデータに関係するトランザクションの総処理量の２～３割程度となる。一方、モジュール間の負荷偏り率ωは１～２割程度であり、方法２を採用した場合、ＳＬＰディレクトリ分のメモリ量は増加するが、モジュール間の負荷を十分に平準化できると考えている。

４.６　結言

カスタマ毎のサービスデータへのトラヒックが数千倍から数万倍と大きくばらつく場合、カスタマ毎のトラヒックを高にものからの順位を用いて近似する方法を提案した。これに基づき、複数のモジュールにサービスデータを分散して配置し、大量のトランザクションを分散処理するシステムを対象として、モジュールに収容するサービスデータ数とモジュール間の負荷の偏りの関係や、その偏りの上限値の推定方法を明らかにした。これらの結果を用い、現状のプロセッサ性能、高度ＩＮサービスにおけるトランザクションのダイナミックステップ数などから、トラヒック量のばらつきが数千から数万倍と大きいサービスデータを無作為に抽出して各モジュールに配置したとしても、モジュール間の負荷の偏りが１～２割程度に収まることを示した。さらに、この偏りに対しては、ＳＬＰディレクトリ検索処理とサービスデータに基づくデータベース処理を効率的に組み合わせることにより、負荷を動的に平準化できることを示した。

モジュール間の負荷の偏りの評価技術は、高度ＩＮのサービス制御ノードに限らず、ＯＬＴＰシステム等、データベースを分散処理するシステムの方式設計や設備設計の効率化に有効である。

なお、本章では、フリーダイヤルサービスのデータに基づいて負荷の偏りについて評価した。しかし、現時点では必ずしもこれらのデータが十分でないため、今後、サービス毎、カスタマ毎のトラヒック量の測定を拡充し、提案した近似式の検証をさらに進めてゆく必要がある。また、どのカスタマについてもトラヒックが集中する時間帯は同一とし、かつこの時間帯についてはトラヒックは一様に分布すると仮定して検討を進めた。しかし、すべてのカスタマについてトラヒックの集中する時間帯が同一になるわけではなく、サービス毎、カスタマ毎の時間変動を含めたトラヒック量の測定も拡充してゆく必要がある。これらについては、今後の課題と考えている。

第５章
分散処理によるサービス制御ノードの
モジュール間結合技術

５．１　緒言

高度インテリジェントネットワーク（高度ＩＮ）のサービス制御ノード（ＳＣＰ）では、小規模なサービスから全国的な大規模サービスまで柔軟に対応することや需要の不透明なサービスへ効率的に対応することが要求される。このような需要の変動にタイムリに対応可能とするためには、多数のモジュールからなる大規模な分散構成によりＳＣＰを実現することが有効である[5]。分散処理システムを拡張性良く高信頼に実現するためには、モジュール間を接続する結合機構のデータ転送能力に制約が生ぜず高い信頼性が達成されることが要求される。また、結合機構の高信頼化に加えて、モジュール個々には高い通信処理能力が要求される。トランザクション処理の場合は、モジュール間で送受されるデータ長は数１００バイトと短いが、送受を高頻度で行う必要がある。これを実現するため、通信処理を専用に行うハードウェア（以下、通信制御チャネルと記す）を適用し、この通信制御チャネルの単位時間当たり処理可能な通信回数を向上させることが重要となる。

本章では、分散構成を採るＳＣＰ内のモジュール間を結合する方式として、十分なデータ転送能力を確保することからスイッチ方式とし、モジュールとのインタフェースにはＡＴＭを適用する分散処理システムを対象として、結合機構の高信頼化構成法や、結合機構に接続されるモジュールの通信処理能力を向上させる制御方式ついて提案する[51]～[54]。はじめに、ＡＴＭ結合機構を各モジュールと接続する機能、信号を多重化／分離化する機能、スイッチングを行う機能により実現し、機能単位に二重化してモジュール数が増加した場合でも信頼度の低下を抑えるとともに、接続するモ

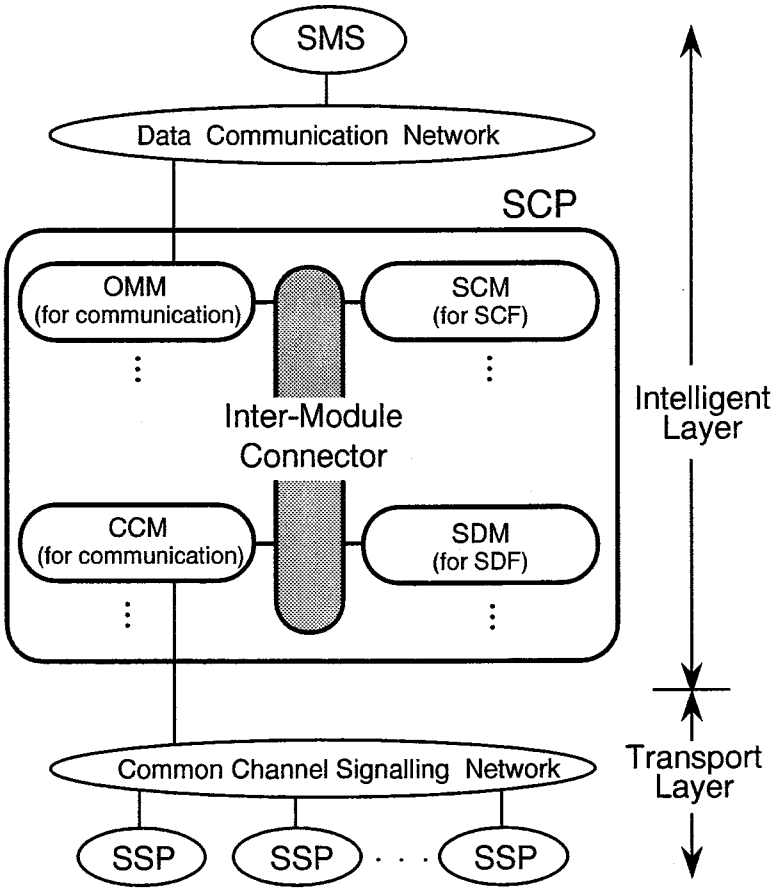
ジュール数に応じて機能単位に増設可能とした、高信頼で経済的な構成法を明らかにする。次に、モジュールの通信処理能力を向上させる制御方式として、ソフトウェアからの通信要求を主メモリにキューイングし、通信制御チャネルが一定周期毎にまとめて処理する方式を取り上げ、周期時間と応答時間の関係を明らかにし、高度ＩＮのＳＣＰでは数１０ミリ秒の周期で処理することが有効であることを示す。さらに、以上の検討結果を踏まえて試作した分散処理システムの評価を行うため、モジュール間での通信に要する時間を測定する性能測定プログラムの構成法や測定結果について示す。

5.2　サービス制御ノード構成条件

5.2.1　ノード構成条件

高度ＩＮに適用するＳＣＰを構成するためには、ＩＴＵ－Ｔで標準化[41]されているサービス制御機能（ＳＣＦ）やサービスデータ機能（ＳＤＦ）に加えて、交換機、サービス管理ノード（ＳＭＳ）との通信機能が必要となる。機能間の分離性がよく、負荷変動に対して機能単位に柔軟に対応できること、実現する機能に最適化した技術を適用できることから、これらの機能を別々のモジュールに配備し、負荷の増大に対しては機能単位で負荷分散するノード構成を想定する[5]。すなわち、ＳＣＦはサービス制御モジュール（ＳＣＭ）、ＳＤＦはサービスデータモジュール（ＳＤＭ）、共通線信号網を介した交換機との通信は通信制御モジュール（ＣＣＭ）、サービス管理ノードとの通信は運転、保守モジュール（ＯＭＭ）で実現し、負荷の増大に対しては各機能を実現するモジュールの追加により対応する。ＳＣＰのハードウェア構成を図5.1に示す。

ＳＣＰの規模としては、パーソナル通信など全国規模のサービス、カスタマオリエンティッドな多様なサービスへの適用を考え、従来ノード[10]に比べ大規模化し、



SMS : Service Management System
SCP : Service Control Point
SSP : Service Switching Point
SDF : Service Data Function
SCF : Service Control Function
SDM : Service Data Module
SCM : Service Control Module
CCM : Communication Control Module
OMM : Operation & Maintenance Module

図５.１　ＳＣＰ構成

１００万カスタマ程度を収容可能とする。また、ＳＣＰにかかるトラヒックは２０００呼／秒程度を想定する。接続されるモジュール数は呼当たりのダイナミックステップ数やプロセッサ性能に依存し、約５０モジュール程度を想定する。

5.2.2　性能条件

交換機に収容される電話機から発信された呼は一般の電話呼かインテリジェントネットワークを利用する高機能な電話サービスを受ける電話呼（以下、ＩＮ呼と記す）かを判別され、ＩＮ呼の場合は共通線信号網を介してトランザクションがＳＣＰに送られる。ＳＣＰはサービス毎、カスタマ毎にどのような内容のサービスを行うかのデータベースを有し、このデータベースの内容に従い、交換機と協調して高機能な電話サービスを実現する。本章で想定する機能分散と負荷分散を統合したＳＣＰでは、交換機から送られてきたトランザクションは、まずＣＣＭで受信され、結合機構を介してＳＣＭへ転送される。ＳＣＭでは必要に応じて、ＳＤＭにアクセスし、データベースの参照および更新を行い、トランザクション処理結果をＣＣＭを経由して交換機に送信する。

１つのＩＮ呼を処理するためには少なくとも発信時と切断時にそれぞれ１回のトランザクションが発生する。また、サービス内容によっては、呼の接続処理中に交換機からＳＣＰへの問合せのトランザクションが必要となることもある。ＩＮ呼当たりのトランザクション数としては２～４ Tr／呼を想定する。また、１つのトランザクションを処理するために必要なモジュール間の通信回数は、サービス内容にも依存するが、５回／Tr 程度である。従って、ＩＮ呼当たりのモジュール間の通信回数は１０～２０回／呼となり、単位時間当たり２０００呼を処理するために必要なモジュール間の総通信回数は２００００～４００００回／秒程度となる。呼の処理のためにモジュール間で送受されるメッセージのデータ長は約２００Ｂ程度であり、モジュール間の総通信量は３０～６０M bps 程度となる。ＳＣＰにかかるトラヒックとモジュール間の通信量の関係を図５.２に示す。

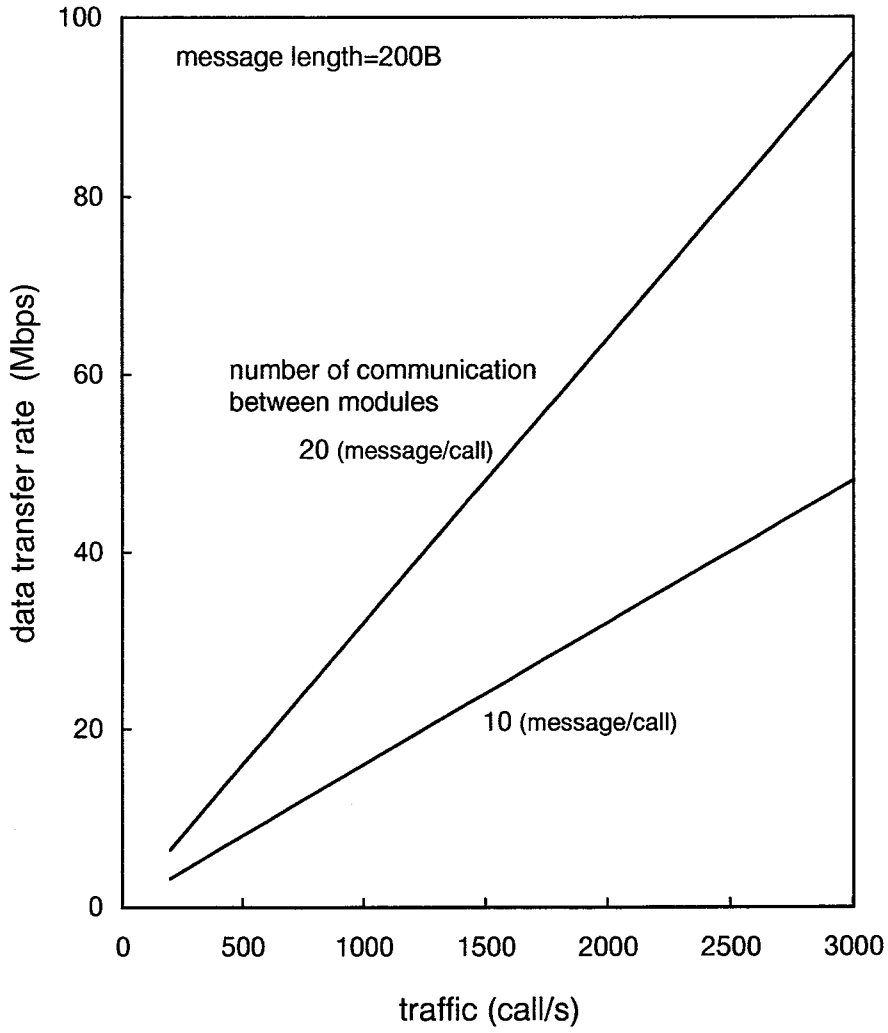


図５.２　トラヒックとモジュール間のデータ転送量

モジュールに要求される通信処理能力は、入出力を合わせると、総通信回数の２倍をモジュール台数５０で割った値となり、１０００～２０００回／秒程度となる。

一般の電話呼の場合、ダイヤル終了から呼出し音が返るまでの時間が接続品質として規定されている。着信先の位置等の条件による変動はあるがおおよそ数秒から１０秒程度である。高度ＩＮサービスの場合は交換機からＳＣＰへの問合せが必要であり、一般の電話呼に比べて、呼出し音が返るまでに要する時間が、この問合わせの分だけ余計にかかる。高度ＩＮサービスの接続品質を考慮すると、ＳＣＰへの問合わせの遅延時間は１秒程度以下に抑える必要がある。この時間より交換機とＳＣＰ間の共通線信号網の遅延時間を差し引き、ＳＣＰノード内で許容される時間は３００ミリ秒程度を想定する。

モジュール間の通信には、呼処理のための通信以外に、ＳＭＳからカスタマのデータをＳＣＰ内のモジュールにダウンロードするための通信や運用情報をＳＭＳに送るための通信等がある。このようなダウンロードや運用のためのトラヒックは、呼処理のためのトラヒックに比べ２～３桁小さいため、モジュール間の総通信量や総通信回数としては、呼処理のための通信のみを考えておけば十分である。このため、本章ではダウンロード等のトラヒックについては対象外とした。

５.２.３　信頼度条件

高度ＩＮサービスは図５.１に示すように交換機の上位にＳＣＰを配置する形で実現され、交換機とＳＣＰのそれぞれの不稼働率が加算されてサービスに影響する。高度ＩＮサービスを一般の電話サービスと同程度の不稼働率でユーザに提供するためにはＳＣＰの不稼働率が交換機の不稼働率に比べてはるかに小さく、無視できる程度であることが望ましい。交換機の信頼度条件としては２０年間に１時間程度のシステムダウン（不稼働率≒6×10^{-6}）が設定されている[55]。本章では、ＳＣＰの不稼働率を交換機の不稼働率より１～２桁低く想定し、10^{-7}～10^{-6}以下（２０年間に１～１０分以下のシステムダウン）を目標とする。ＳＣＰを構成するモジュールは相互バッ

クアップ等により信頼度上無視できる程度とすることができる。分散処理システムの信頼度のボトルネックはモジュール間の結合機構であり、結合機構の不稼働率の目標は10^{-7}～10^{-6}以下とする。

5.3　モジュール間結合方式

5.3.1　モジュール間結合方式の選択

モジュール間の結合方式は、大きくバス／リング結合とスイッチ結合に大別できる。ＳＣＰのモジュール間の結合方式としては、以下の理由によりスイッチ結合とし、モジュールとスイッチ間はＡＴＭにより接続することとした。

ＯＬＴＰシステム等で用いられているバス／リング結合の代表的なものとしてＦＤＤＩがある。ＦＤＤＩの最大データ転送速度は１００ Mbps であるが、実効データ転送能力は電文長が短くなるに従って小さくなり[56]、２００Ｂ程度の短電文では１０～３０％程度である。ＳＣＰとしての所要データ転送能力（３０～６０ Mbps）を確保するためには、複数のＦＤＤＩリンクを実現することが必要となり、負荷の分散や障害時のソフトウェア処理が複雑となる。また、ＦＤＤＩリンクを制御する通信制御チャネルをモジュール毎に複数実装する必要がありコストアップにもつながる。一方、スイッチ結合の場合は高いデータ転送能力を実現することは容易である。たとえば、１入力当たり１５６ Mbps の転送能力を有する８×８のＡＴＭスイッチの場合、１.２ Gbps のデータ転送能力があり、転送能力に対する制限はほとんど無いと考えて良い。

また、モジュール間の通信をＡＴＭで実現しておくと、将来、共通線信号網を介したＳＳＰとの通信や情報転送網を介したＳＭＳとの通信等のノード間の通信がＡＴＭ化された場合、モジュール間の通信とノード間の通信を同様に扱えるメリットがある。

5.3.2　ＡＴＭ結合機構の高信頼化構成法

ＦＤＤＩはモジュール間で通信を行うためのスイッチング機構が各モジュールに分散されており、１台のモジュール内のスイッチング機構が障害になったとしてもＳＣＰノード全体の障害とはならない。一方、スイッチ結合の場合は、スイッチング機構が一カ所に集中するため、結合機構の障害が即座にＳＣＰノード全体の障害につながる。このため、結合機構の高信頼化が必須となる。

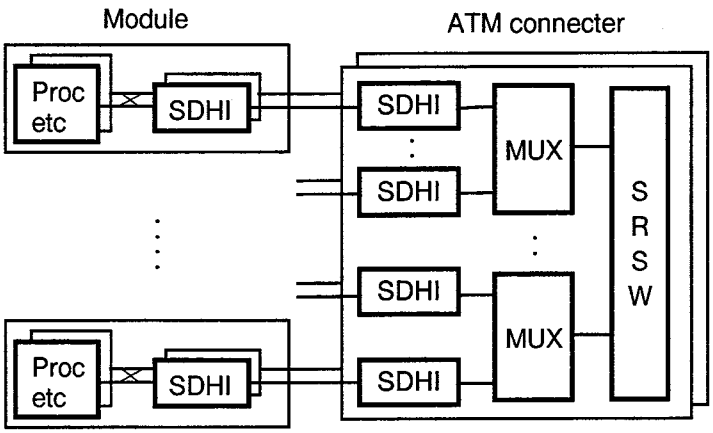
ＡＴＭ結合機構は各モジュールと接続する光インタフェース部（ＳＤＨＩ）、各モジュールからの／への信号を多重化／分離化する部分（ＭＵＸ）、ＡＴＭセルのルーチングを行うスイッチ部（ＳＲＳＷ）の機能ブロックから構成することができる。ＳＤＨＩは接続するモジュール毎に必要であり、モジュール台数Ｎに比例して数が増加する。ＭＵＸは信号を多重化するモジュール数ｍに依存し、モジュール台数がＮの場合、ＭＵＸの台数Ｋは次式で表せる。

$$K=[(N-1)/m]+1 \qquad (5.1)$$

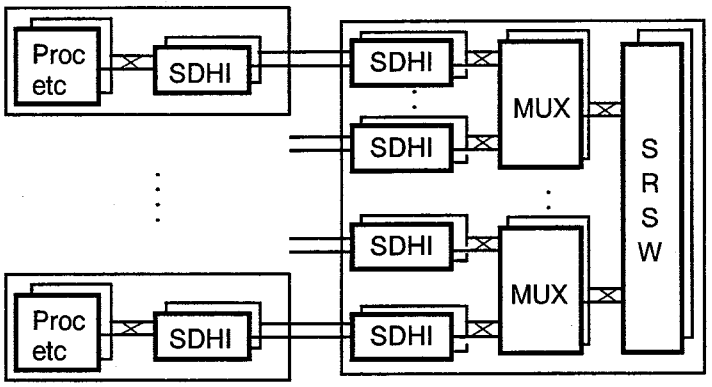
ＳＲＳＷの増設は難しく、あらかじめ最大のノード規模に見合ったものを１台用意する必要がある。

ＡＴＭ結合機構の高信頼化をはかるためには各機能ブロックを二重化する必要があるが、二重化の方法として、ＡＴＭ結合機構全体として二重化する方法（方法１）とＡＴＭ結合機構を構成する機能ブロック毎に二重化し、機能ブロック間で交差を設ける方法（方法２）が考えられる。両方法のＡＴＭ結合機構の構成を図５.３に示す。ＡＴＭ結合機構に接続するモジュールはプロセッサとＳＤＨＩから構成され、それぞれ二重化されているとする。また、プロセッサとＳＤＨＩ間には交差が設けられており、現用系のプロセッサが障害となり予備系に切替わったとしてもＳＤＨＩやＡＴＭ結合機構の切替えは不要とする。

方法１は多数のパソコン等を接続した企業内のＬＡＮ等で幅広く使用されている一重化構成のＡＴＭ結合機構を２台用いて全体として信頼度を向上させる方法に相当する。方法１では現用系のＳＤＨＩ、ＭＵＸ、ＳＲＳＷのどれか１台が故障した場



Method 1



Method 2

SDHI : Synchronized Digital Hierarchy Interface
MUX : Multiplexing and Demultiplexing
SRSW: Self-Routing Switch

図５.３　ＡＴＭ結合機構の信頼度構成

合でもＡＴＭ結合機構の切替えが生ずる。また、モジュール側のＳＤＨＩの故障に対してもＡＴＭ結合機構の切替えが必要となる。方式２の場合は、機能ブロック間で交差が設けられており、現用系の、ある機能ブロックが故障した場合は、当該機能ブロックのみ予備系に切替えることとなる。方法１、２の信頼度構成モデルを図５．４に示す。

方法１、２の不稼働率をそれぞれＵＡ$_1$、ＵＡ$_2$とおくと、次式で表せる。

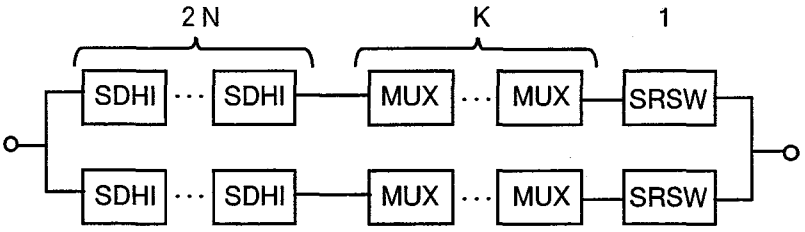
$$UA_1 = \frac{\lambda_1^{\,2}}{(\mu + \lambda_1^{\,2})} \tag{5.2}$$

$$UA_2 = \frac{N(2\lambda_{SDHI})^2}{(\mu + 2\lambda_{SDHI})^2} + \frac{K\lambda_{MUX}^2}{(\mu + \lambda_{MUX})^2} + \frac{\lambda_{SRSW}^2}{(\mu + \lambda_{SRSW})^2} \tag{5.3}$$

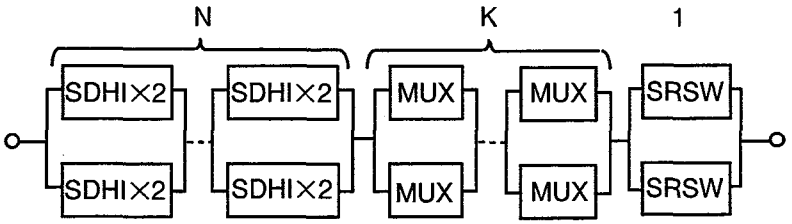
但し、　$\lambda_1 = 2N\lambda_{SDHI} + K\lambda_{MUX} + \lambda_{SRSW}$　(5.4)

である。ここで、λ_{SDHI}、λ_{MUX}、λ_{SRSW}はそれぞれＳＤＨＩ、ＭＵＸ、ＳＲＳＷの故障率、μは修理率である。モジュール台数と不稼働率の関係を図５．５に示す。図５．５において、試作したハードウェア量からλ_{SDHI}、λ_{MUX}、λ_{SRSW}はそれぞれ2.5×10^{-5}、5×10^{-5}、10^{-4}とした。また、従来の交換機の修理状況を考慮しμは１とした。方法１は図５．５からわかるように接続するモジュール台数の増加とともに不稼働率が増加し、要求される信頼度条件（10^{-7}～10^{-6}以下）のもとでは、接続可能なモジュール数は数台～１０数台に制限され、大規模ノードを構成することは難しい。一方、方法２は機能ブロック間で交差を設けるため、モジュール台数が増加してもＡＴＭ結合機構の不稼働率の増加は小さく、５０台程度のモジュールを接続した大規模システムにおいても要求される不稼働率を満足する。

方法１、方法２ともに結合機構を構成する機能ブロックは同一であり、機能ブロックのハードウェア量の差はほとんどない。しかし、方法２の場合は０系と１系の機能ブロック間に交差を設けるため、機能ブロック間を結合するケーブル類が多く必要となる。このため、結合機構のコストは方法２が方法１に比べて若干高くなる。しかし、システム全体で見るとモジュールのコストが支配的であり、結合機構のコスト差に基づくシステムコストの差はほとんどない。また、結合機構のスループットやレスポンスタイムの性能については方法１と方法２で差はない。



Method 1



Method 2

図５．４　ＡＴＭ結合機構の信頼度ブロック図

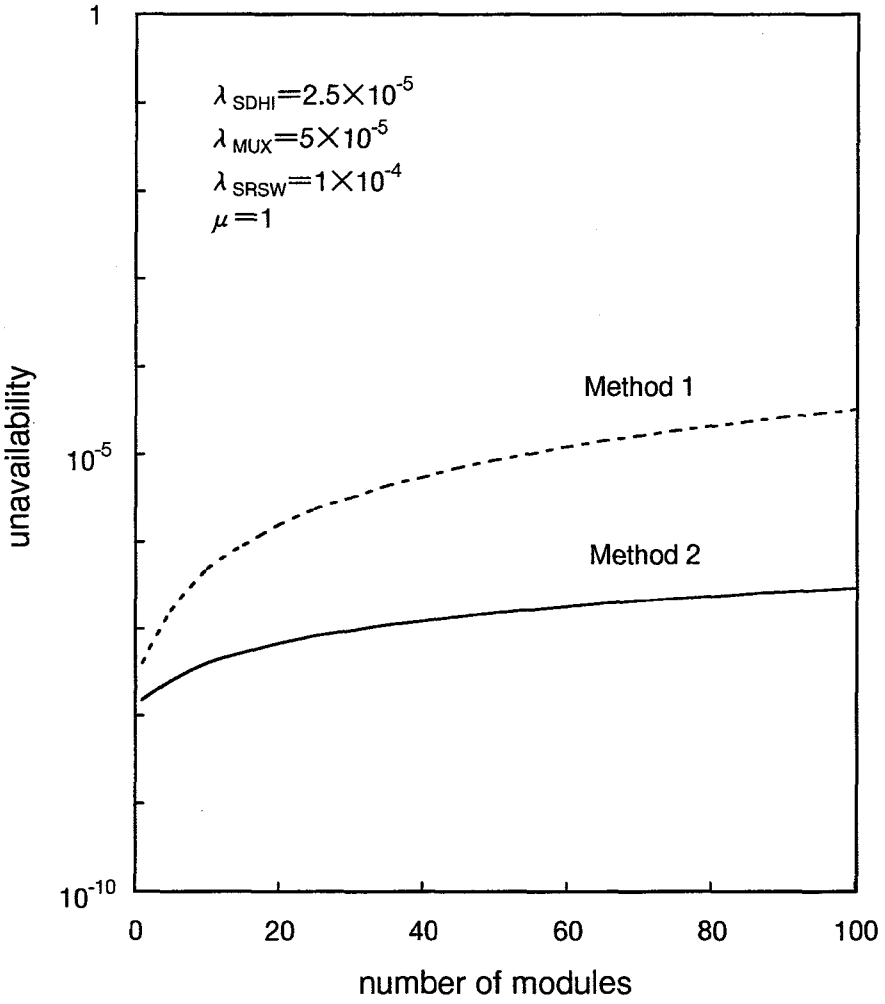


図５.５　ＡＴＭ結合機構の不稼働率

以上、信頼度面では方法２は大規模構成においても要求される信頼度条件を満足するが、方法１はモジュール数が１０数台以上になると要求条件を満足しない。一方、コスト面では方法１が方法２に比べて若干有利である。一般に規模が大きくなるに従ってシステムとして要求される信頼度条件は高くなる[57]。このことから、ＡＴＭ結合機構としては、機能単位に交差を設け、接続するモジュール台数が増加した場合でも信頼度の低下度合いが小さい方法２を採用することとした。

方法２において、機能ブロック間の交差の実現方法を図５.６に示す。図５.６はＳＤＨＩとＭＵＸ間の交差の例を示しており、実線で囲った機能ブロックは現用系、破線で囲った機能ブロックは予備系を示す。現用系、予備系のＳＤＨＩから信号が両系のＭＵＸに供給され、ＭＵＸでは両方の信号のどちらか一方を選択する回路（図５.６のＳ）を有し、通常は現用系のＳＤＨＩからの信号を選択する。現用系のＳＤＨＩが故障した場合には予備系のＳＤＨＩの信号へ切替える。また、ＭＵＸ内に両系のＳＤＨＩからの信号の途絶や同期ずれを検出する監視回路（モニタ）を設け、現用系のＳＤＨＩからの信号に異常を検出した場合にはモニタが自律的に選択回路を予備系のＳＤＨＩからの信号に切替える。図５.６には示していないが、ＭＵＸとＳＲＳＷ間の交差も同様の方法で実現した。

５.３.３　ＡＴＭ結合機構の全体構成

高度ＩＮサービスは小規模から大規模まで幅広く分布する。このため、要求に応じてモジュールを追加し、柔軟に処理能力を拡張できることが要求される。併せて、接続するモジュール数が少ない場合、結合機構のコスト負担が少ないことが要求される。このことから、ＡＴＭ結合機構を機能ブロック単位に増設可能とし、ＳＤＨＩについては１モジュール単位に、ＭＵＸについては８モジュール単位に、接続するモジュール台数に応じて増設可能とした。

システムを立上げる場合は、ＡＴＭ結合機構を構成する機能ブロックの初期化やルーチング情報の設定を行う必要がある。また、モジュールを増設した場合にもルーチ

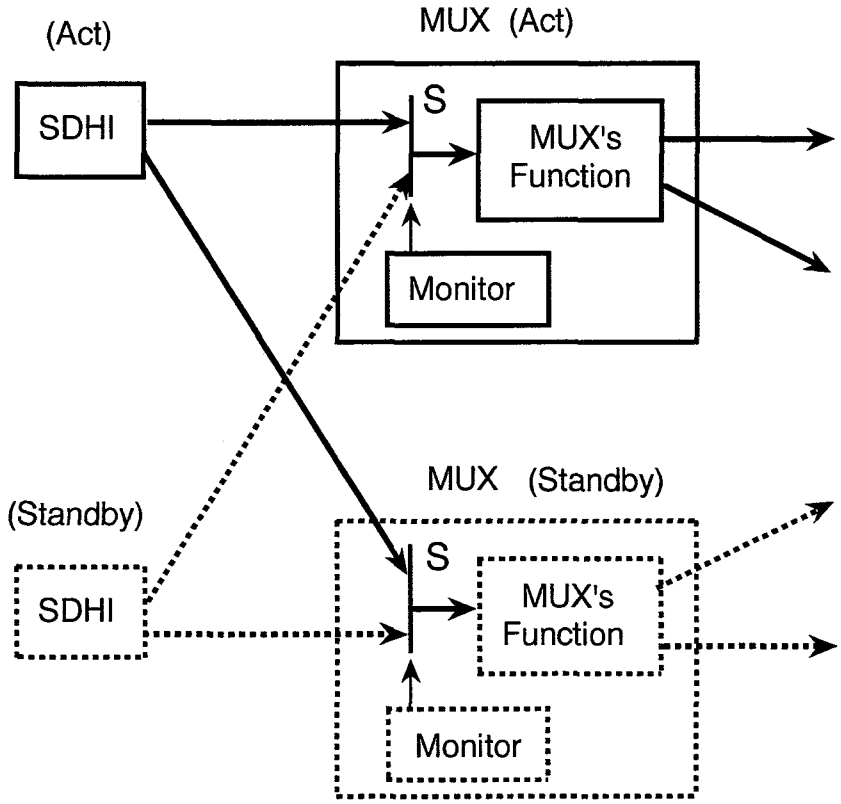


図５．６　機能ブロック間の交差の実現法

ング情報の再設定が必要となる。このような処理は一度行えば変更されることは無いため、通常の呼処理でプロセッサに負荷がかかることはない。このため、これらの処理を行うプロセッサは、ＳＭＳとの通信を行うモジュール（ＯＭＭ）と共用することとし、ＡＴＭ結合機構をこのモジュールと一体化し、システムインタフェースモジュール（ＳＩＭ）として実現した。ＡＴＭ結合機構およびプロセッサを含むＳＩＭの全体構成を図５.７に示す。

ＳＩＭの実装に関しては、一つの架の中にプロセッサやＳＲＳＷ、ＭＵＸ、ＳＤＨＩを搭載し、この架で２４モジュールを接続可能とした。また、より多くのモジュールを接続可能とするため、別の架にＭＵＸやＳＤＨＩを搭載し、最大で８８モジュールまで接続可能とした。ＳＩＭの実装構成を図５.８に示す。各モジュールとＡＴＭ結合機構とのインタフェース速度は、２００Ｂ程度のメッセージを１０００～２０００回／秒の頻度で送受した場合でも十分対応可能とするため６ Mbps とした。

５.４　通信制御方式

ＡＴＭ結合機構に接続され、モジュールの通信処理を専用に行う通信制御チャネルの制御方式や応答時間について考察する。

５.４.１　制御方式

通信制御チャネルの制御方式として即時処理と周期処理がある。即時処理とは、ソフトウェアでデータ転送の要求が発生すると同時に通信制御チャネルに起動をかけ、通信制御チャネルでは起動内容に応じた処理を即座に実行し、処理終了と同時にソフトウェアへ割込みにより通知する方式である。応答時間条件の厳しい磁気ディスク装置等を接続する入出力チャネルでは一般に即時処理が採用されている。一方、周期処理とは、ソフトウェアでデータ転送の要求が発生すると、その転送要求をメモリ上の

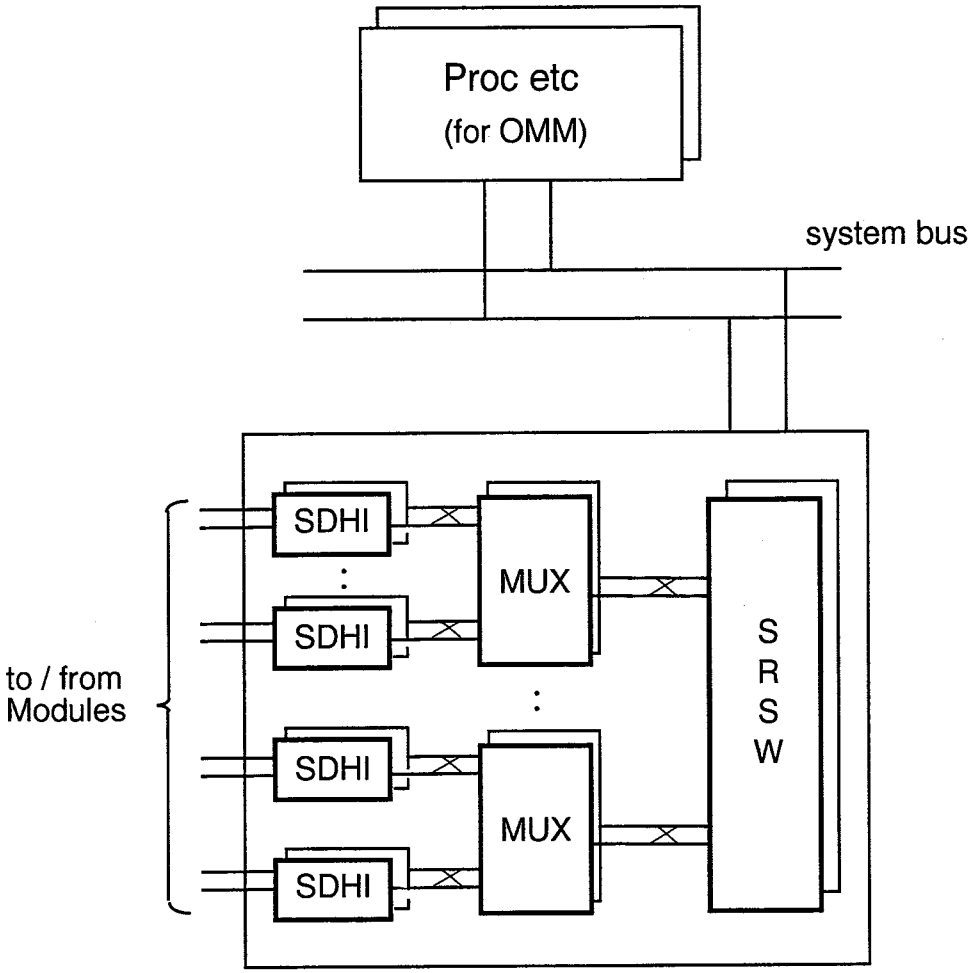


図５．７　ＳＩＭの全体構成

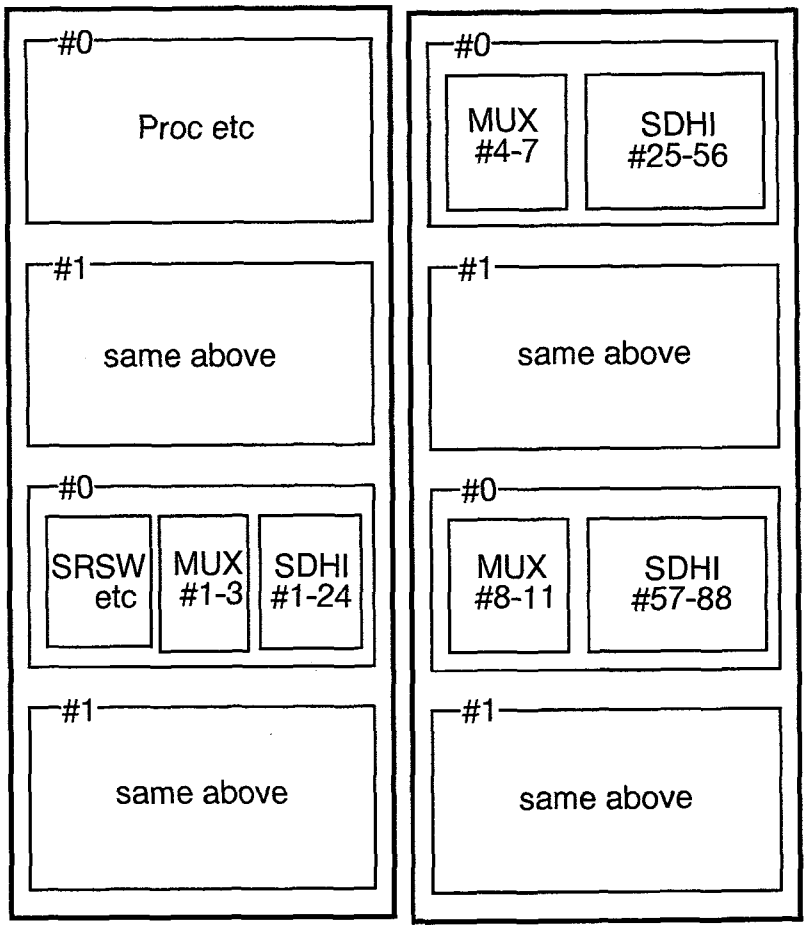


図５．８　ＳＩＭの実装構成

キューに格納するのみで、通信制御チャネルには起動をかけない方式である。通信制御チャネルは、あらかじめ決められた周期でメモリ上のキューをルックインし、処理要求の有無を調べ、それに基づいて処理する。処理が終了するとメモリ上のキューにその旨を書き込む。ソフトウェアから非同期に要求される起動処理やソフトウェアへの終了割込み処理が不要となること、メモリ上のキューから複数の転送要求を読出すことが可能であることから、スループットの向上が期待できる。本章で対象とするモジュールの通信制御チャネルには１０００～２０００回／秒の高い通信処理能力が必要とされるため、通信制御チャネルの制御方式として、周期処理を採用することとした。周期処理の場合、周期時間を長く設定すると一度に多くの通信要求を処理できるためスループットは向上するが、転送要求がメモリ上のキューに書込まれてからルックインするまでは待たされるため、ソフトウェアからの転送要求から通信制御チャネルでの処理が終了するまでの時間（以下、応答時間と記す）が長くなることが懸念される。周期時間の設定に当たっては応答時間の評価が必要となる。これに関しては、まず、ＳＣＰに要求されるトランザクション処理時間から通信制御チャネルの応答時間への要求条件を明らかにし、次に簡単な単一キュー、単一窓口の待ちモデルにより、周期時間と応答時間の関係を以下で評価する。

５．４．２　通信制御チャネルの応答時間条件

ＳＣＰノード内でのトランザクション処理時間T_{RES}は、プロセッサの処理時間T_{CPU}、モジュール間の通信時間T_{COM}、トランザクション当たりのモジュール間通信回数ｆにより、次式で表せる。

$$T_{RES} = T_{CPU} + f \cdot T_{COM}$$
$$= \frac{D}{(1-\rho)P} + f \cdot (2T_{CCH} + T_{ATM}) \qquad (5.5)$$

ここで、Ｄはトランザクション当たりのダイナミックステップ数、Ｐはプロセッサの性能、ρはプロセッサの使用率、T_{CCH}は通信制御チャネルの応答時間、T_{ATM}はメッセージがＡＴＭ結合機構を通過する時間である。なお、モジュール間の通信に要する

時間T_{COM}は図5.9に示すように、送信モジュール側の通信制御チャネルの応答時間、メッセージがＡＴＭ結合機構を通過する時間、受信モジュール側の通信制御チャネルの応答時間の和で表せるとした。また、通信制御チャネルの応答時間は、送信、受信で同一とした。

5.2節で述べたように、T_{RES}に許容される時間は約３００ミリ秒程度である。ρを８０%、Ｄを１００ｋステップ、Ｐを２０ MIPS とすると、T_{CPU}は２５ミリ秒となる。T_{ATM}はメッセージのデータ長（約２００Ｂ）、モジュールとＡＴＭ結合機構とのインタフェース速度（6 Mbps）で決まり１ミリ秒程度である。トランザクション当たりのモジュール間通信回数ｆは５回／Trであり、T_{CCH}に許容される時間は２５ミリ秒程度となる。

5.4.3　評価モデル

通信制御チャネルへの通信要求はポアソン到着とし、窓口は一定周期Ｔ毎に開き、キューから周期時間内に処理可能なメッセージ数ｂを取り出して処理する。キュー内のメッセージ数が周期時間内に処理可能なメッセージ数より少ない場合は、キューからすべてのメッセージを取り出して処理する。このとき、取り出した全てのメッセージの処理が終了したとしても周期途中で再度窓口を開いて、新たに発生したメッセージを処理することはしない。キュー内のメッセージ数が周期時間内に処理可能なメッセージ数より多い場合はｂ個のメッセージのみ処理し、残りは次の周期で処理する。周期Ｔ内に発生したメッセージ数をｎ、処理残りのメッセージ数をｈとすると、周期の終了時点ではキューの中にｈ＋ｎ個のメッセージがある。評価モデルを図５．１０に示す。

5.4.4　通信制御チャネルの応答時間評価

周期の始めに行うキュー内のメッセージの検索等周期内で処理するメッセージの

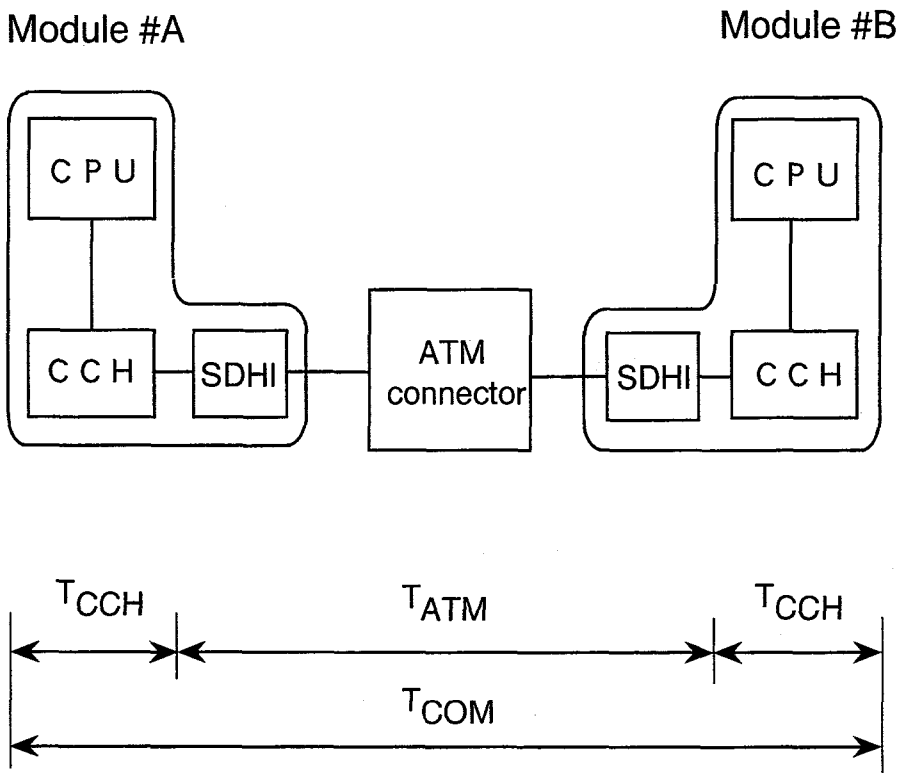


図５．９　モジュール間の通信時間

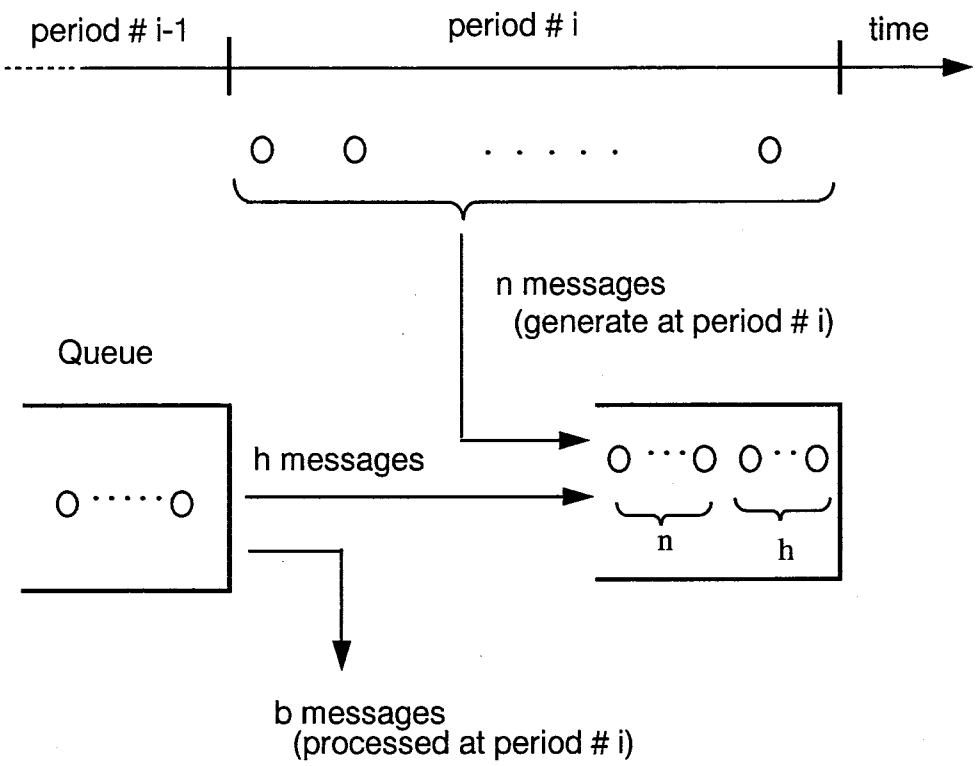


図５．１０　周期駆動モデル

数に依存せず必要な固定分の処理時間をｔ$_0$、メッセージの処理に必要な処理時間をｔ$_1$とおくと、周期Ｔ内に処理可能なメッセージ数ｂは

$$b=\frac{T-t_0}{t_1} \tag{5.6}$$

となる。固定分ｔ$_0$を周期内で処理するメッセージで一様に分担するとしたときのメッセージの平均処理時間ｓは

$$s=\frac{T}{b}=t_1+\frac{t_0\cdot t_1}{T-t_0} \tag{5.7}$$

となる。以下では簡単化のため、周期Ｔ内においてメッセージは平均処理時間ｓで処理されるものとして評価する。Ｔを大きくすることによりｓは小さくなりスループットは向上するが、発生したメッセージは周期終了まで待たされるため応答時間は長くなる。ｎ個のメッセージについて、発生から処理が終了するまでの平均時間（以下、平均応答時間と記す）を$t_{CCH}(n,h)$とする。ｎ個のメッセージは周期の終了時点まで待たされ、次の周期では、まず、処理残りのｈ個のメッセージの処理が終了した後、キューに登録された順に処理されるとする。平均応答時間$t_{CCH}(n,h)$はメッセージの発生時点から周期終了までの平均待ち時間Ｔ$_q$、処理残りのｈ個のメッセージの処理時間Ｔ$_h$、ｎ個のメッセージの平均処理終了時間Ｔ$_n$の和となり、次式で表せる。

$$t_{CCH}(n,h)=T_q+T_h+T_n \tag{5.8}$$

ここで、ｎ個のメッセージは周期Ｔ内でランダムに発生するためＴ$_q$はＴ／２、１つのメッセージの処理時間はｓでありＴ$_h$はｓ・ｈとなる。また、ｎ個のメッセージのうち、１番目のメッセージの処理が終了する時間はｓ、２番目のメッセージの処理が終了する時間は２ｓ、ｎ番目のメッセージの処理が終了する時間はｎ・ｓであり、平均処理終了時間Ｔ$_n$は（ｎ＋１）ｓ／２となる。従って、$t_{CCH}(n,h)$は次式となる。

$$t_{CCH}(n,h)=\frac{T}{2}+s\cdot h+\frac{(n+1)s}{2} \tag{5.9}$$

周期内でｎ呼発生する確率をＰ（ｎ）、処理残り呼数がｈである確率をＸ（ｈ）とおくと、すべてのメッセージの平均応答時間Ｔ$_{CCH}$は次式で表せる。

$$T_{CCH} = \sum_{h=0}^{\infty}\sum_{n=0}^{\infty}\{n \cdot X(h) \cdot P(n) \cdot t_{CCH}(h,n)\} / \sum_{n=0}^{\infty}\{n \cdot P(n)\} \tag{5.10}$$

式（5.10）に式（5.9）を代入し次式を得る。

$$T_{CCH} = \frac{T}{2} + s \cdot \overline{X} + \frac{s}{2a} \cdot \overline{P} \tag{5.11}$$

ここで、 $$\overline{P} = 1 \cdot 2 \cdot P(1) + 2 \cdot 3 \cdot P(2) + 3 \cdot 4 \cdot P(3) + \cdots \tag{5.12}$$

$$\overline{X} = X(1) + 2X(2) + 3X(3) + \cdots \tag{5.13}$$

である。単位時間当たりの平均メッセージ数をＡとおくと、周期Ｔ内で発生する平均メッセージ数ａおよびＰ（ｎ）は次式で表せる。なお、Ｘ（ｈ）の算出方法は付録５．１に示す。

$$a = A \cdot T \tag{5.14}$$

$$P(n) = \frac{e^{-a} \cdot a^{n}}{n!} \tag{5.15}$$

ＡとＴ$_{CCH}$の関係を図５．１１に示す。図５．１１において、試作した通信制御チャネルで走行するファームウェアのダイナミックステップ数等からｔ$_{0}$は０．５ミリ秒、ｔ$_{1}$は０．５ミリ秒とした。図５．１１より、周期Ｔを長くすることにより、Ａは向上するが、向上度合いはＴが長くなるに従って飽和する。たとえば、Ｔが１６ミリ秒以上ではＡはほぼ飽和する。一方、Ｔ$_{CCH}$はＴが長くなるに従って長くなる。Ｔが１６ミリ秒では、Ａが１８００（呼／秒）でＴ$_{CCH}$は約２０ミリ秒程度であり、Ａをこれ以上大きくするとＴ$_{CCH}$は急激に増加する。

以上の検討結果から、通信制御チャネルの制御方式として、１６ミリ秒で周期処理することとした。このとき式（5.5）より、モジュール間の通信時間Ｔ$_{COM}$は４０ミリ秒程度、トランザクション処理時間Ｔ$_{RES}$は２２５ミリ秒程度となり、５．２節で述べた要求条件（３００ミリ秒）を満足する。

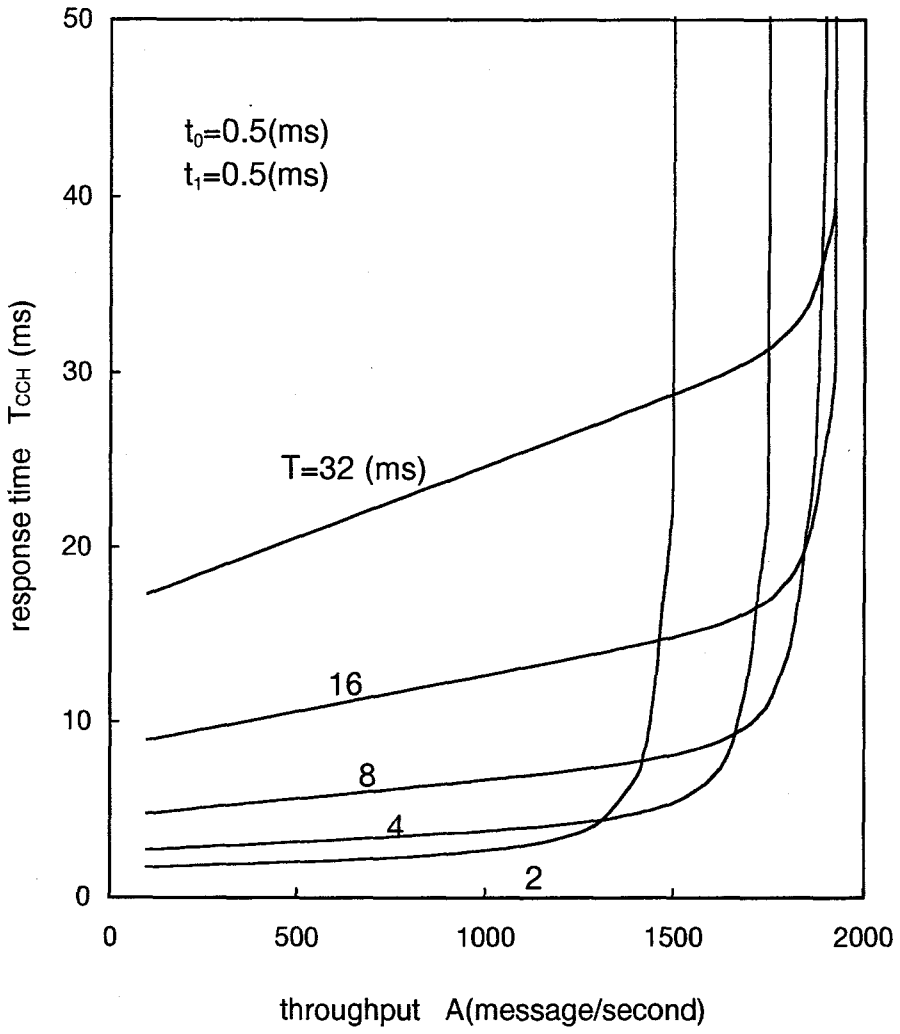


図５．１１　スループットと応答時間

5.5　評価方法

本節では、前節での検討結果の検証を目的として作成したモジュール間の通信時間を測定する性能測定プログラムの構成法および測定結果について考察する。

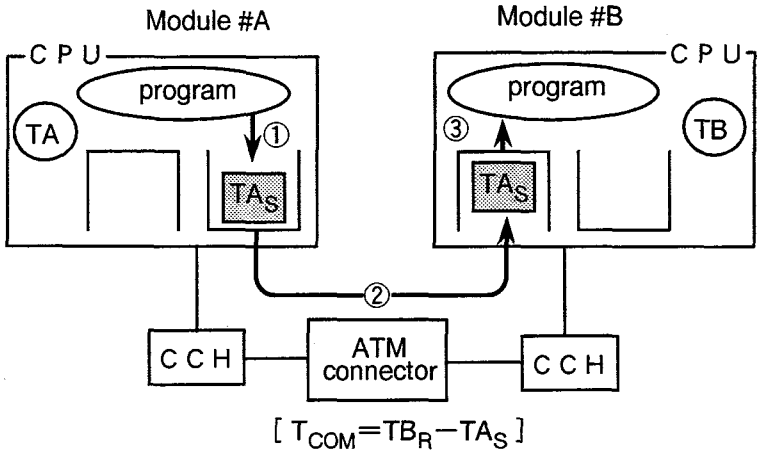
5.5.1　性能測定方法

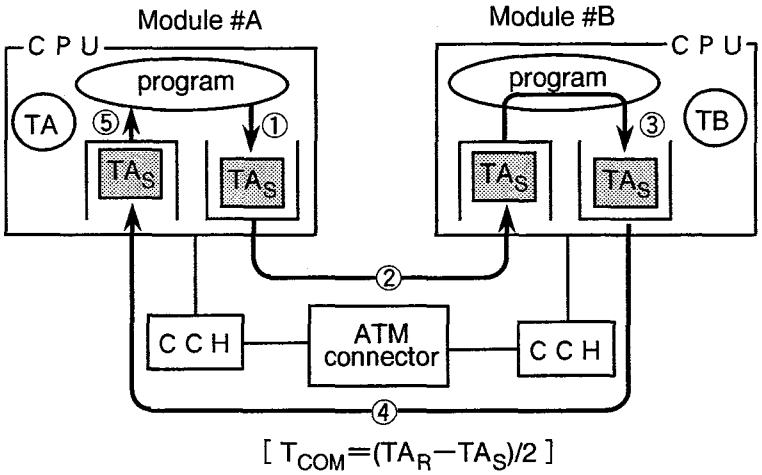
試作した装置上で走行するプログラムによりモジュール間でメッセージの送受信を行い、通信に要する時間T_{COM}を実測評価する。通信時間の測定方法として以下の2案が考えられる。

【方法Ⅰ】モジュールAでキューに送信コマンドを登録するとき、メッセージのデータ部に登録時刻を書き込む。このメッセージがモジュールBで受信された時刻と、メッセージに埋め込まれた時刻の差によって通信時間を測定する。

【方法Ⅱ】モジュールAでキューに送信コマンドを登録するとき、メッセージのデータ部に登録時刻を書き込む。当該メッセージがモジュールBで受信されると、当該メッセージのアドレスをモジュールA宛に変更して送り返す。モジュールAでメッセージを受信した時刻とメッセージに埋め込まれた時刻との差により通信時間を測定する。この場合、往復の時間をカウントしたこととなり実際のモジュール間の通信時間はこの値の1／2となる。

方法Ⅰ、Ⅱの測定方法を図5.12に示す。方法Ⅰの場合は、2つのモジュールのタイマを利用して通信時間を測定するため、タイマ値の差により誤差が生ずる。モジュール間でのメッセージの通信に要する時間は数10ミリ秒程度であり、2つのモジュールのタイマ値をこれに見合った精度で一致させることは非常に難しい。一方、方法Ⅱの場合は、1つのモジュールのタイマを利用して通信時間を測定するため、高い





TAS : Message from Module #A to Module #B

: Queue on main memory

TA : Timer in Module #A

TB : Timer in Module #B

T_{COM} : Communication time between Modules

TA_S : Time of TA when the message was sent by Module #A

TB_R : Time of TB when the message was received by Module #B

TA_R : Time of TA when the message was received by Module #A

図５．１２　モジュール間の通信時間測定方法

精度で測定できる。また、モジュール間でタイマ値に差が生じたとしても測定に誤差は生じない。このことから、本性能測定プログラムでは方法Ⅱを採用することとした。

5.5.2　性能測定プログラムの走行方法

性能測定プログラムを各モジュール上で走行させるためには、各々のモジュールに対して、プログラムのローディングや走行指示が必要となる。また、各モジュール上の性能測定プログラムを同期をとって走行させる必要がある。方法として以下の２案が考えられる。

【方法Ａ】モジュール毎のパネルやコンソールから行う方法

【方法Ｂ】ワークステーションと各モジュールを接続し、ワークステーションから行う方法

方法Ａはモジュール毎に人を配置する必要があり、モジュール数が少ない場合はよいが、モジュールの増加とともに、操作が大変となる。また、モジュール毎に細かい同期を取ることが難しくなる。方法Ｂはワークステーション上で新たなプログラムを開発する必要があるが、モジュールが増加しても一人で走行させることが可能であり操作性がよい。また、モジュール間の同期を取ることも容易にできる。本章では、モジュール数が多い大規模ノード構成時でも一人で操作できることを考慮し、方法Ｂを採用することとした。ワークステーション上のプログラム規模としては、１０キロライン程度で実現できた。各モジュールとワークステーションはイーサネットで接続し、一台のワークステーションから性能測定プログラムのローディング、起動および測定結果の表示等を行うこととした。性能測定プログラムの走行環境を図５.１３に示す。

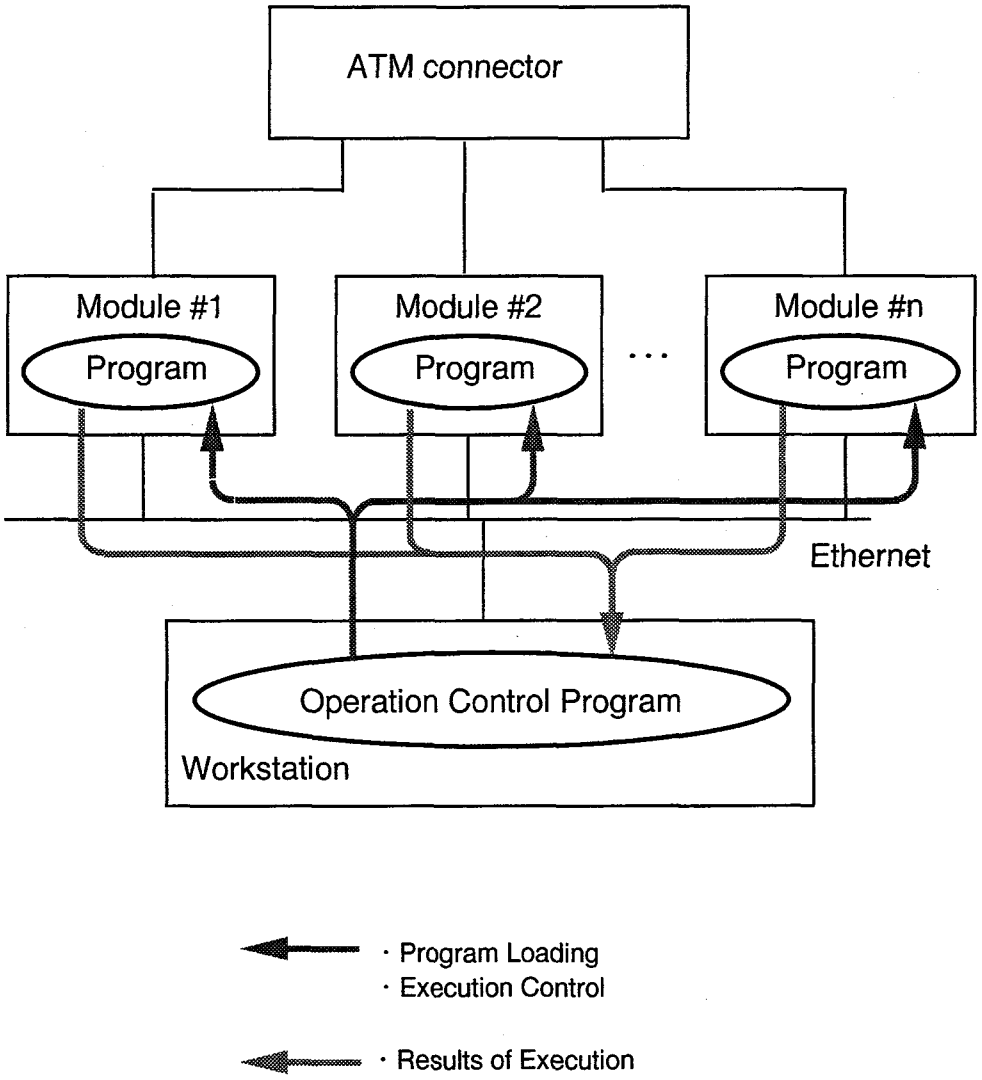


図５．１３　性能測定プログラムの走行環境

5.5.3　性能測定結果

性能測定プログラムを試作した装置上で走行させて実測した通信時間の分布を図5.14に示す。図5.14では転送するデータ量を振らせて測定した。図5.14よりT_{COM}の平均は約40ミリ秒であり、ほぼ5.4節での評価通りの値となった。また、データ長が256Bでも90%以上の通信は50ミリ秒以内に収まることが判明した。データ転送量が大きくなるに従って通信時間が長くなる理由はATM結合機構の通過時間や、5.4節では考慮しなかったが通信チャネルと主メモリ間のデータ転送時間がデータ長に比例して増加するためである。

5.6　結言

大規模分散処理システムの結合機構の高信頼化構成法、通信処理能力を向上させる通信制御チャネルの制御方式およびモジュール間の通信時間の測定方法について提案した。モジュール間の接続にATMを適用し、結合機構を各モジュールと接続する機能、信号を多重化／分離化する機能、スイッチングを行う機能により実現し、機能単位に二重化した高信頼化構成法を明らかにした。これにより、結合機構に50モジュールを接続した大規模分散構成においても、不稼働率が10^{-7}～10^{-6}の高い信頼性を実現できた。また、各モジュールの通信制御については、ある周期毎にまとめて複数のメッセージを処理する方式を取り上げ、周期時間と通信時間の関係を明らかにし、高度INのSCPでは数10ミリ秒の周期で処理することが有効であることを示した。ここで示したモジュール間の結合方式は、高信頼で経済的な大規模分散処理システムを設計する上で有用である。本章では、高度INに適用するサービス制御ノードを対象として検討を進めたが、ここで提案した方式は、適用業務の拡大から多数のモジュールによる分散処理化が進んでいるOLTPシステムにおいても、同様に適用できる。

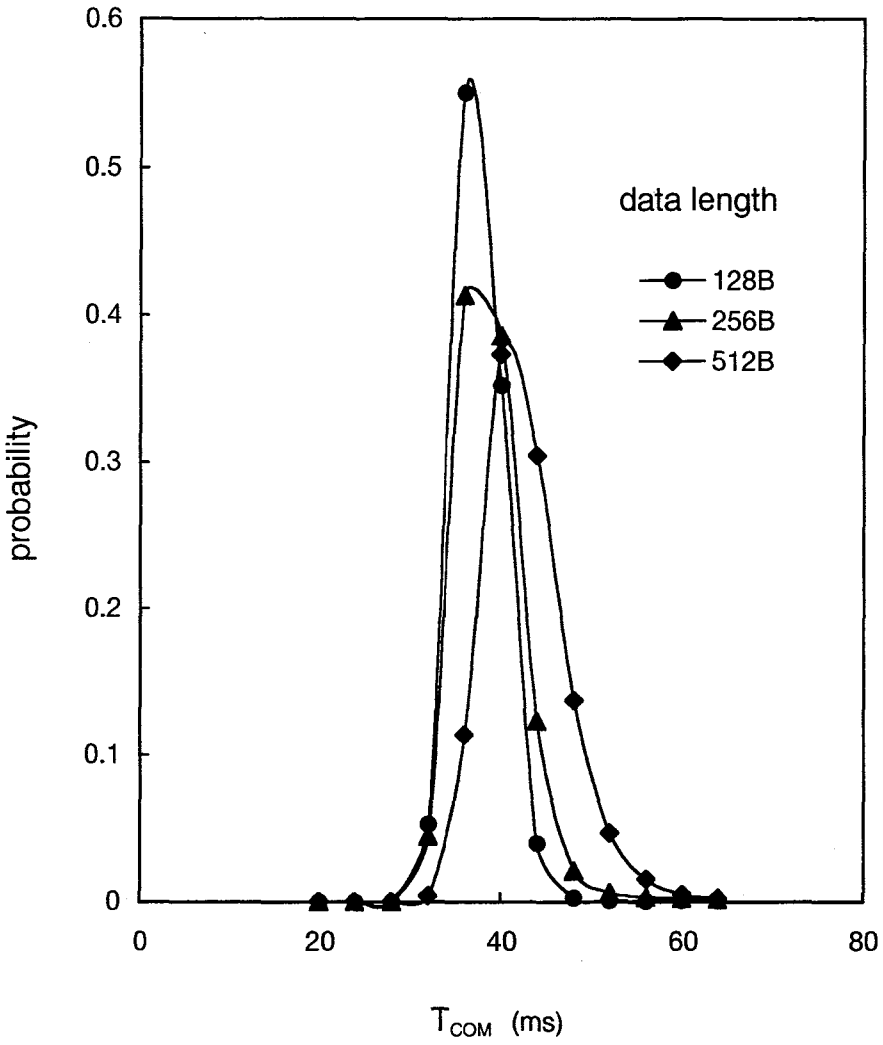


図５．１４　モジュール間の通信時間の分布

第６章
サービス制御ノードにおける
メモリデータベースのリカバリ技術

６．１　緒言

高度ＩＮのＳＣＰでは、多数のカスタマの情報から構成されるデータベースをもとに高度な通信サービスを実現するため、データベースを効率よく高速に処理できることが重要であるが、同時に障害に対しても、これらデータベースの内容が十分な確度で保証されることが要求される。また、高度ＩＮのデータベースは容量が比較的小さく、アクセス頻度が高い特徴を有し、トランザクション処理性能の向上のためには、第３章で述べたようにデータベースのすべてを主メモリ上に常駐するメモリデータベースの適用が有効である。

既に、銀行のオンラインシステム、クレジット照会システム等のＯＬＴＰ（On-Line Transacsion Processing）システムでもサービスの広域化、対象業務の拡大を背景に、より大量のトランザクションを高速に処理する必要がでてきており、主メモリの大容量化の進展と相まって、メモリデータベース化が進められている。しかし、メモリデータベースは、ソフトウェアのバグ、ハードウェア障害および電源障害により、主メモリ上のデータベースが破壊、喪失されやすいという欠点を併せ持っている。このため、あらかじめリカバリに必要な情報を障害に強い記憶装置上に格納しておき、障害時にはこの記憶装置上のリカバリ情報をもとに障害直前のデータベースを復旧する方式を併せて採用する必要がある[5][28][29]。外部の記憶装置の形態はＬＯＧを単に記憶するパッシブな方式とＬＯＧをプロセッサで処理可能なアクティブな方式に２大別される。前者を代表し高速な入出力が可能な半導体ディスク装置を適用する方式を設定し、後者については別のモジュールのプロセッサに付属する主メモリを適用する方式を代替案として設定する。後者の方式では、当該の主メモリ上でＬＯＧを逐一データベースに反映する方法をとると、リカバリ時間の大幅な短縮が期待できる。

本章では、２つの代替案を中心に、データベースへの単位時間、容量当たりのトランザクション数やリカバリ時間、モジュール毎に収容可能なデータベース容量に注目し、両方式の適用領域を明らかにする。また、電源障害時の不揮発化方法、冗長化構成法など信頼度構成面から考察し、サービス制御ノードのメモリデータベースのリカバリ方式としては、バッテリバックアップによる半導体ディスク装置を用いたリカバリ方式が有効であることを示す。ここで用いるバッテリバックアップについては、試作を通じて、保守の容易な方法を明らかにする等、全体として高度ＩＮで実用性の高いリカバリ方式を提案する[58]～[62]。ここで示したリカバリ方式はメモリデータベース化が進められている多くのＯＬＴＰシステムにおいても有用と考えている。

６.２　前提条件

高度ＩＮサービスの特性に基づいて、本章で前提とした条件は以下の通りである。

（１）高度ＩＮのデータベースは容量が小さくアクセス頻度が極めて高い。パーソナル通信サービスを例にとると、カスタマ対応のデータとしては位置情報、暗証番号、スクリーニング情報等からなり数１００バイト程度である。トランザクションとしてはカスタマからの発信、カスタマへの着信及び着信先を指定する位置登録などがあり１時間当たり３回程度、すなわち、カスタマ当たりの平均トランザクション数は10^{-3}（Tr/秒）程度であり、データベースへの単位時間、容量当たりのトランザクション数（以下、トランザクション密度と記す）は１～１０（Tr/MB/秒）程度となる。データベースのサイズはモジュールに収容するカスタマ数に依存し、プロセッサ性能やトランザクション当たりのダイナミックステップ数等によってきまる。収容するカスタマ数を数１０万程度とし、データベース容量としては１００ＭＢ程度を想定する。

（2）一般の電話呼の場合、ダイヤル終了から呼出し音が返るまでの時間が接続品質として規定されている。着信先の位置等の条件による変動はあるがおおよそ数秒から１０秒程度である。高度ＩＮサービスの場合は交換機からＳＣＰへの問合せが必要であり、一般の電話呼に比べて、呼出し音が返るまでに要する時間が、この問合わせの分だけ余計にかかる。高度ＩＮサービスの接続品質を考慮すると、ＳＣＰへの問合わせの遅延時間は１秒程度以下に抑える必要がある。この時間より交換機とＳＣＰ間の共通線信号網の遅延時間を差し引くとＳＣＰノード内で許容される時間は数１００ミリ秒、データベース処理に許容される時間は１００ミリ秒程度である。

（3）高度ＩＮでトランザクション処理中に更新されるデータとしてはパーソナル通信サービスの位置情報、回線が使用中か否かを表示するフラグなどがある。前者はいったん登録されたデータが失われると、その後の呼の接続が出来なくなりユーザサービスに大きな支障が出るため、ＬＯＧ取得による情報の高信頼化が必要となる。一方、後者はリカバリ時に初期値に戻すことでユーザサービスに支障をきたさず、網内で対処できるデータで、ＬＯＧの取得は必ずしも必要でない。

（4）データベースのリカバリを行っている間はサービスが一時中断し、ユーザからの呼を受付けることができない。通常、ユーザは再度ダイヤルすることとなるが、１回の試行に１０～２０秒程度かかる。ユーザの再試行を考慮すると、数秒以内での高速なリカバリは必ずしも必要とされない。また、リカバリ時間が数分以上になると再試行の呼と新たな呼が加算されて輻輳を引き起こすことが懸念される。さらに、当初、呼が受付けられなかったユーザも数回ダイヤルした時点でサービスが再開できていた場合には、ほとんど障害を意識することはない。このことを考慮するとリカバリに許容される時間は６０秒程度となる。

（5）ＳＣＰの構成としては、データベースを複数のモジュールに分割して配備し、トランザクションを分散処理する負荷分散を前提とした。１モジュールとしての能力は、高度ＩＮのトランザクション処理のダイナミックステップ数および最近のプロセッサの性能から、単位時間当たり処理可能なトランザクション数は数１００（Tr/秒）とする。また、地震、火災などの災害に対しても十分な信頼度を確保するため、ノード内ではモジュールを単位とした二重化を行い、ノード間では２つのノード相互のバックアップを行う。呼処理は交換機とその上位に配置したＳＣＰで行う。サービスオーダ処理やサービス内容の変更等のカスタマ対応はＳＣＰの上位に配置されるＳＭＳで行う。すなわち、カスタマ情報等の管理はＳＭＳで行い、呼処理に必要な情報のみがＳＭＳからＳＣＰへダウンロードされる。

（6）高度ＩＮのノードにおいても交換機と同等の保守で対応可能なことが必須である。交換機に用いている論理回路から構成されるパッケージの保守はモジュールからの故障の報告にもとずき予備パッケージと交換して正常性を確認することにより行われる。電源障害に対応するため半導体メモリのバックアップにバッテリを用いたとしても、定期保守を不要とし、論理回路から構成されるパッケージと同等の保守条件で対応可能とする。

6.3　評価モデルの設定

前節で述べたように単位時間当たり数１００のトランザクションを処理するためには、ＬＯＧを取得する外部の記憶装置は数１００（回/秒）の入出力が要求される。最近の磁気ディスク装置の単位時間当たりの入出力回数は４０～６０（回/秒）程度でありＬＯＧ取得への適用は難しい。トランザクション毎にＬＯＧを取得するためには、外部の記憶装置に高い入出力能力が要求され、半導体メモリの使用が必須となる。磁

気ディスク装置を複数並べて並列動作させるアレーディスク装置については、入出力回数の面では適用の可能性もあるが、容量面での使用効率が極端に低下するため、本章では対象とはしない（付録６．１参照）。

高頻度のＬＯＧ取得が可能な外部記憶系の構成として、当該モジュールのプロセッサ配下に接続する半導体ディスク装置と、別のモジュールのプロセッサに付属する主メモリとを代替案として、次に示す方式１、２を設定する。プロセッサ外部へのデータ送受はファイル装置への入出力処理による方法とプロセッサ間通信による方法の２通りに大別されるが、両方式はこれに対応している。加えて、方式２では単にＬＯＧを取得するだけでなく、本来のトランザクション処理と並行してＬＯＧに関係する処理を行えるなど、この面からもリカバリ時間短縮の可能性を併せて評価することができる。なお、両方式とも半導体メモリとしてＤＲＡＭを使用することとした。このため、電源障害時にデータが消失する問題があり、バッテリによるバックアップが必要となるが、これに対しても併せて考察する。

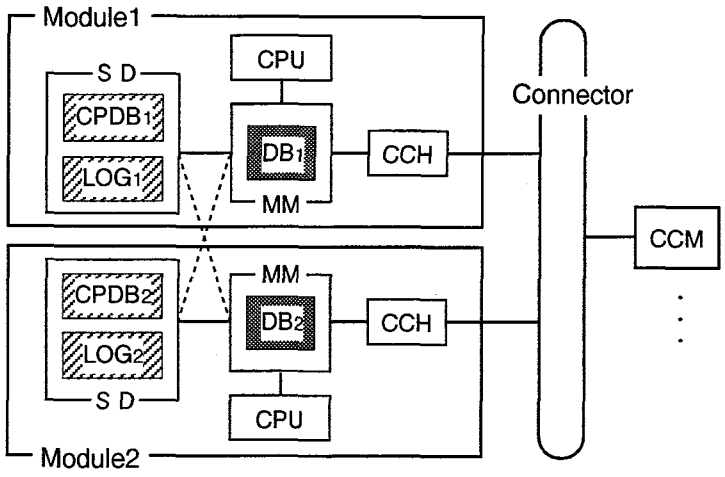
方式１：リカバリ情報の格納媒体として半導体ディスク装置を用いる。本方式ではトランザクション毎にＬＯＧを半導体ディスク装置に書込む。また、リカバリ時間を短縮するため、トランザクション処理と並行して一定周期でＣＰＤＢを半導体ディスク装置に取得する。なお、取得したＬＯＧを長時間に渡って保持することは不要であり、リカバリに必要なＣＰＤＢの書込みが終了した場合は、ＣＰＤＢ取得開始時点より前のＬＯＧは破棄する。

方式２：リカバリ情報の格納媒体として他のモジュールの主メモリを用いる。トランザクション毎に通信チャネルを介して他のモジュールにＬＯＧを送る。他のモジュールではデータベースのコピー（ＤＢＣ）を有し、受信したＬＯＧをプロセッサによりＤＢＣに上書きすることによって、常に最新のデータベースを保持する。このため、ＣＰＤＢは使用しない。なお、ＤＢＣに上書き後、ＬＯＧは破棄する。

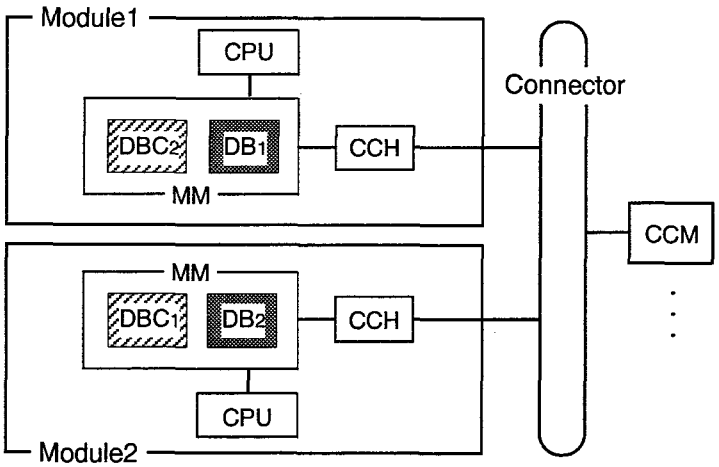
高度ＩＮではＬＯＧを長時間に渡って保持しておくことは不要であり、本章ではリカバリに不必要となった後はＬＯＧを破棄するとした。ＬＯＧを長時間に渡って保持する必要がある場合は、方式１ではＳＤ上のＬＯＧを破棄する前に大容量の記憶媒体に移す方法、方式２では相互にバックアップ関係にある双方のモジュールで、ＬＯＧを主メモリ上である程度バッファリングして大容量の記憶媒体に移す方法が考えられる。いずれにしてもＬＯＧをバッファリングする必要があり、本章の方式１、２をベースに、ＬＯＧを大容量の記憶媒体へ書き込む処理を追加することにより容易に拡張できる。

方式１、２共にモジュールが障害となった場合、他のモジュールで処理を再開するための予備を含めたシステム構成例を図６.１に示す。２台のモジュールが１／２づつ負荷を分担し相互にバックアップする。ＯＳやＤＢＭＳ等のプログラムファイルの格納媒体としては磁気ディスク装置を使用するが、プログラムファイルはシステム起動時のみにアクセスされ、呼処理でのアクセスはない。このため、本章で対象とするメモリデータベースのリカバリ方式の選定に直接影響しないことから図６.１では省略している。

トランザクションの受け付けから応答までのプロセッサ（ＣＰＵ）、半導体ディスク装置（ＳＤ）、通信チャネル（ＣＣＨ）での処理の流れを図６.２に示す。 方式１についてはＣＰＤＢ取得処理のフローについても併せて示す。図６.２の直線上の記号はＣＰＵ、ＳＤ、ＣＣＨの処理時間である。なお、１回の入出力に要するＳＤの処理時間、ＣＣＨの処理時間は、アクセス時間とデータ長に比例するデータ転送時間との和で表わしている。ここで、図６.２で使用している記号の内容は表６.１に示す通りである。



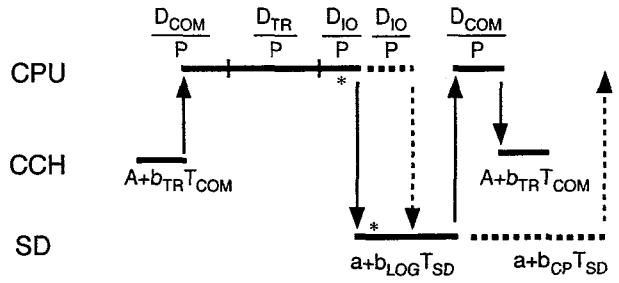
(1) Method 1



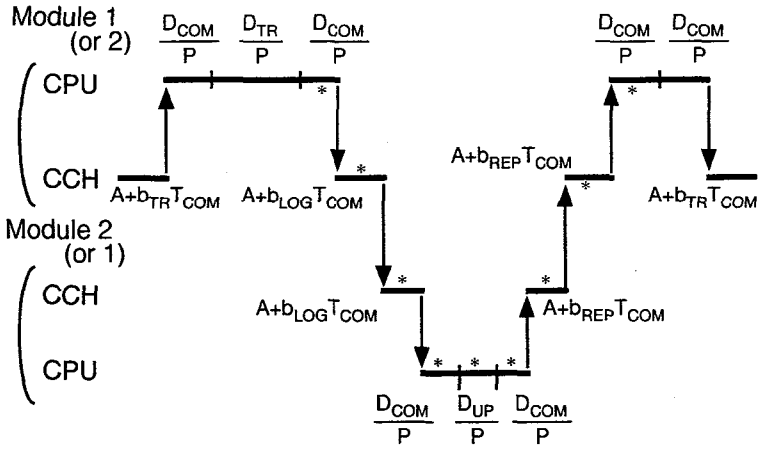
(2) Method 2

DB: Current database
LOG: Log data
MM: Main memory
CCH: Communication channel
CPDB: Database at the checkpoint
DBC: Database copy
SD: Semiconductor disk device
CCM: Communication Control Module

図6.1　システム構成



(1) Method 1



(2) Method 2

▬ : Transaction Processing ▪▪▪▪ : Checkpoint Processing

* : Transaction without logging don't need these processing.

図6.2　処理フロー

表６．１　　性能評価パラメータ

分類	記号	内容	設定値
ＤＳ	D_{TR}	トランザクション処理のＤＳ	80KS
	D_{IO}	入出力処理のＤＳ	5KS
	D_{COM}	通信処理のＤＳ	10KS
	D_{UP}	ＬＯＧをＤＢに上書きする処理のＤＳ	10KS
データ長	b_{LOG}	ＬＯＧのデータ長	512B
	b_{TR}	トランザクションのデータ長	100B
	b_{REP}	方式２でバックアップ側モジュールからのＬＯＧ受信の応答のデータ長	100B
	b_{RCV}	リカバリ時に１回の入力で半導体ディスク装置から読込むデータ長	64KB
	b_{CP}	ＣＰＤＢ取得のため，１回の出力で半導体ディスク装置に書込むデータ長	64KB
装置性能	P	プロセッサ性能	20MIPS
	a	半導体ディスク装置のアクセス時間	1ms
	A	通信チャネルのアクセス時間	0.5ms
	T_{SD}	半導体ディスク装置のデータ転送時間	200ns/B
	T_{COM}	通信チャネルのデータ転送時間	1000ns/B

6.4　性能評価

方式１、２を性能面から比較評価する。

6.4.1　スループット、レスポンスタイム

方式１、２のＣＰＵ使用率ρ_{CPU1}、ρ_{CPU2}はそれぞれ次式で表せる。

$$\rho_{CPU1} = (\lambda D_1 + \frac{MD_{IO}}{b_{CP}T_{CP}})/P \tag{6.1}$$

$$\rho_{CPU2} = \lambda D_2 / P \tag{6.2}$$

ここで、λは単位時間当たりに当該モジュールで処理するトランザクション数、T_{CP}はＣＰＤＢの取得周期、Ｍはデータベース容量である。D_1、D_2は方式１、２のトランザクション処理に必要なダイナミックステップ数（ＤＳ）であり次式となる。ωはＬＯＧ取得が必要なトランザクションの割合である。

$$D_1 = D_{TR} + 2D_{COM} + \omega D_{IO} \tag{6.3}$$

$$D_2 = D_{TR} + 2D_{COM} + \omega(4D_{COM} + D_{UP}) \tag{6.4}$$

ＣＰＵの最大使用率をρ^M_{CPU}とすると、ＣＰＵの能力に基づく、方式１、２の単位時間当たり処理可能なトランザクション数（以下、スループットと記す）λ_{CPU1}、λ_{CPU2}はそれぞれ次式で表せる。

$$\lambda_{CPU1} = \frac{\rho^M_{CPU}P - MD_{IO}/b_{CP}T_{CP}}{D_1} \tag{6.5}$$

$$\lambda_{CPU2} = \frac{\rho^M_{CPU}P}{D_2} \tag{6.6}$$

方式１の場合、ＬＯＧ、ＣＰＤＢをＳＤに書き込むため、ＳＤの使用率ρ_{SD1}は次式で表せる。

$$\rho_{SD1} = \lambda\omega(a + b_{LOG}T_{SD}) + \frac{M}{b_{CP}T_{CP}}(a + b_{CP}T_{SD}) \tag{6.7}$$

ＳＤの最大使用率をρ^M_{SD}とおくと、ＳＤの能力に基づくスループットλ_{SD1}は次式と

なる。

$$\lambda_{SD1}=\frac{\rho_{SD}^{M}-\dfrac{M}{b_{CP}T_{CP}}(a+b_{CP}T_{SD})}{\omega(a+b_{LOG}T_{SD})} \tag{6.8}$$

方式１の場合、λ_{CPU1}とλ_{SD1}のどちらか小さい方でモジュールとしてのスループットλ$_1$が決まる。方式２の場合は、λ_{CPU2}によってモジュールとしてのスループットλ$_2$が決まる。スループットを決定する要因としてはＣＣＨのボトルネックも想定される。しかし、ＣＣＨの複数設置による負荷の分散は容易であり、本章ではボトルネックの要因としてＣＣＨは考慮していない。

方式１はトランザクション処理と並行してＣＰＤＢを取得する必要があり、λ$_1$はＴ$_{CP}$に影響される。図６.３にＴ$_{CP}$とλ$_1$の関係を示す。図６.３の算出に用いたＤＳ、装置性能等を表６.１に示す。なお、ωは０.７、ρ_{CPU}^{M}、ρ_{SD}^{M}は０.８とした。Ｔ$_{CP}$の短い領域ではＳＤのボトルネックが発生し、Ｔ$_{CP}$はある値より短くできない。Ｍが１００ＭＢの場合、Ｔ$_{CP}$が３０～４０秒ではＳＤにより、４０秒以上ではＣＰＵによりほぼλ$_1$が決まる。

方式１、２でトランザクション間の競合による待ちが無い場合の処理は図６.２中の実線で示した処理フローに従って進められる。トランザクション処理時間T_{RES1}^{0}、T_{RES2}^{0}はＣＰＵ、ＣＣＨ、ＳＤの処理時間の総和となり次式で表せる。

$$T_{RES1}^{0}=(D_{TR}+D_{IO}+2D_{COM})/P+(a+b_{LOG}T_{SD})+(2A+2b_{TR}T_{COM}) \tag{6.9}$$

$$T_{RES2}^{0}=(D_{TR}+D_{UP}+6D_{COM})/P+\{6A+2(b_{TR}+b_{LOG}+b_{REP})T_{COM}\} \tag{6.10}$$

表６.１の条件では、T_{RES1}^{0}は約８ミリ秒、T_{RES2}^{0}は約１２ミリ秒となる。ポアソン到着、指数分布サービスを想定し、使用率を０.８程度とすると、煩雑になるため式は省略するが、レスポンスタイムは競合がない場合の処理時間の約５倍程度となる。トランザクション間の競合を考慮しても、両方式のレスポンスタイムは１００ミリ秒以内に収まり高度ＩＮの条件を満足する。

しかし、方式１の場合、ＣＰＤＢ取得処理との競合によりトランザクション処理が待たされることも考慮する必要がある。ＣＰＤＢ取得処理のｂ$_{CP}$を大きく設定すると

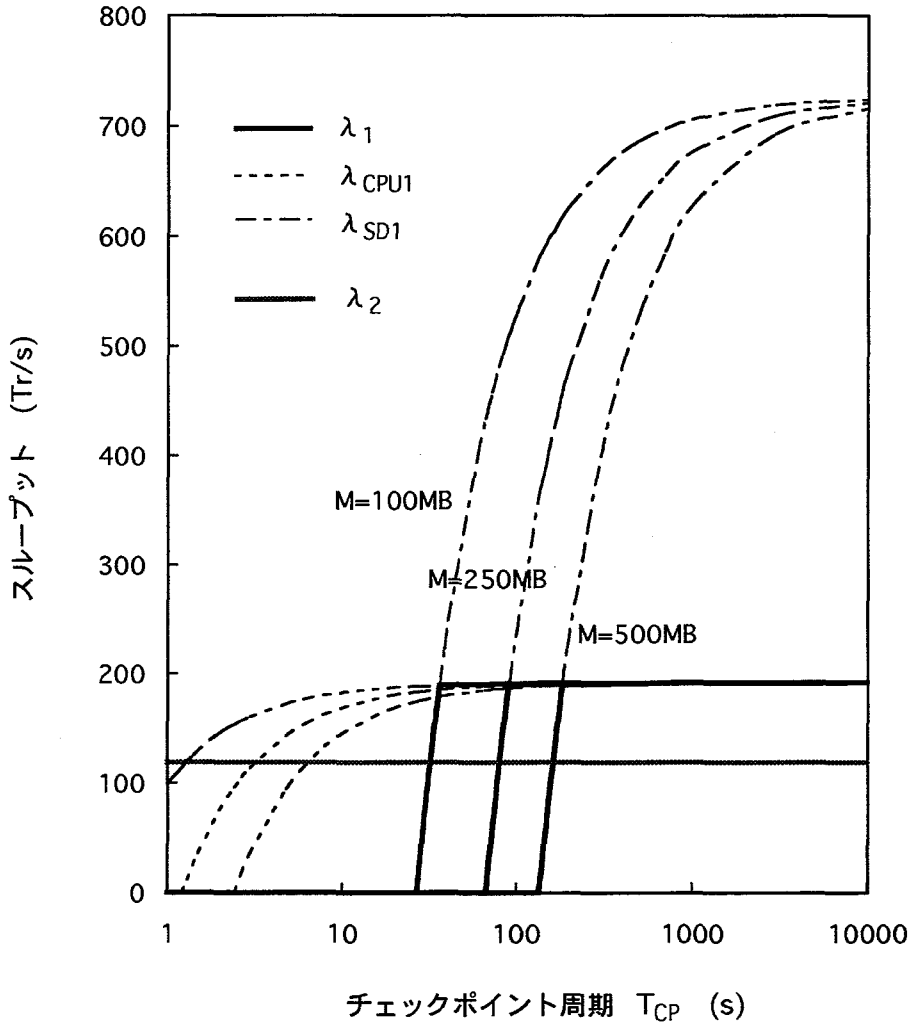


図6.3　チェックポイント時間とスループット

ＳＤの占有時間が長くなり、ＣＰＤＢ取得処理と競合したトランザクションは長く待たされる。これについては、b_{CP}を１００ＫＢ以下に設定すればレスポンスタイムへの影響が無いことを確認した（付録６.２参照）。なお、b_{CP}を小さくするとＣＰＵの負荷が増加するため、本章では６４ＫＢに設定した。

以上、全体としてレスポンスタイムについては方式１、２の間で有意な差は生じないが、スループットに注目すると方式１が優れている。たとえば、Ｍを１００ＭＢ、T_{CP}を６０秒とすると、λ_1はλ_2に比べて約３０％高い。これは一般に、方式２での通信処理は入出力処理に比べてＤＳが大きいこと、ＬＯＧをＤＢＣにその都度反映する処理はＣＰＤＢを取得する処理に比べて、T_{CP}が短くない限り、トランザクション当たりのＤＳが大きいことによる。

６.４.２　リカバリ時間

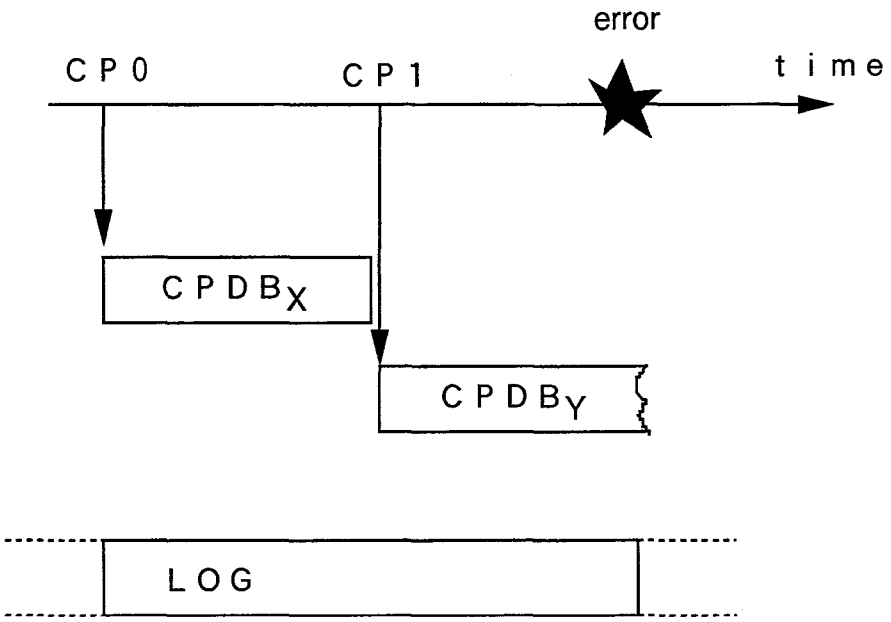
方式１の場合のリカバリ時間T_{RCV1}は、ＳＤからＣＰＤＢと最大で$2T_{CP}$分のＬＯＧを読出す時間と、ＬＯＧをＣＰＤＢに上書きする時間の和で、次式となる。

$$T_{RCV1}=\frac{M+2b_{LOG}\omega\lambda T_{CP}}{b_{RCV}}(a+b_{RCV}T_{SD})+2\omega\lambda T_{CP}\frac{D_{UP}}{P} \qquad (6.11)$$

ここで、$2T_{CP}$とした理由は以下のリカバリを行うためである。

ＣＰＤＢを取得中にエラーが発生した場合、ＣＰＤＢは書換え途中であり、内容が保証されない。このため、ＣＰＤＢの書込みエリアをＳＤ上に２面設け、交互に書込むこととした。ＣＰＤＢ取得方法を図６.４に示す。リカバリを行う場合は、ＣＰＤＢの取得が完了した最新のＣＰＤＢを用いる。図では$CPDB_Y$の取得中にエラーが発生し、$CPDB_Y$が保証できないため、$CPDB_X$からリカバリする。ＬＯＧはＣＰ０から障害時点までのすべてのＬＯＧを読出す。ＬＯＧの量は障害発生ケースに影響されるが、確実なリカバリを保証するため最大値（$2T_{CP}$分のＬＯＧ）とした。

方式２の場合は、既に、他のモジュールの主メモリ上に最新のデータベースが存在するため、リカバリ時間T_{RCV2}は、障害でダウンしたことの検出とサービスを再開す



ＣＰＤＢ：チェックポイントデータベース

→取得エリアを半導体ディスク上に２面（Ｘ，Ｙ）持ち，
交互にＬＯＧを取得する．

ＬＯＧ：データベースの更新ログ

→シーケンシャルに書き込む

図６．４　ＣＰＤＢ取得方法

る手続き等からなり数秒ですむ。

方式１の場合、Ｍを１００ＭＢ、T_{CP}を１００秒、λを１００（Tr/秒）とするとT_{RCV1}は約３０秒となる。リカバリ時間は明らかに方式２が有利である。

６．４．３　データベース容量評価

前節で述べたように方式１はスループットで優れ、方式２はリカバリ時間で優れるという特徴があるため、その優劣を直接判断しにくい。本節ではリカバリ時間を条件としたとき、プロセッサ性能を同一としたモジュールで収容可能なデータベース（ＤＢ）容量を求め、これにより両方式の優劣を総合的に比較評価する。

６．４．３．１　収容可能なデータベース容量

カスタマ対応のデータ量をＲとし、カスタマのデータを使用するトランザクションの単位時間当たりの平均件数をλ_Cとおくと、トランザクション密度ϕは次式で表せる。

$$\phi = \frac{\lambda_C}{R} \tag{6.12}$$

モジュールに収容するカスタマ数をＫとおくと、データベース容量ＭはＫ･Ｒとなり、単位時間当たりのトランザクション数λは、

$$\lambda = \lambda_C K = \frac{\lambda_C}{R} KR = \phi M \tag{6.13}$$

となる。ここで、現状の主メモリの最大容量は数ＧＢ程度であり、λが数１００（Tr/秒）のとき、ϕが約１／１０（Tr/MB/秒）を下まわると容量面からＨＤを使用しなければならず、メモリデータベースに適さない。

方式１で収容可能な最大データベース容量をM_1としたとき、T_{RCV1}は式（6.1）、（6.7）を式（6.11）に代入し、また、λにϕM_1を代入することにより、次式で表せる。

$$T_{RCV1} \geq \frac{M_1(a+b_{RCV}T_{SD})}{b_{RCV}}\left\{1+\frac{2b_{LOG}\omega\phi D_{IO}M_1}{b_{CP}(\rho_{CPU}^{M}P-\phi D_1 M_1)}\right\}+\frac{2\omega\phi D_{UP}D_{IO}M_1^2}{b_{CP}P(\rho_{CPU}^{M}P-\phi D_1 M_1)} \tag{6.14}$$

$$T_{RCV1} \geq \frac{M_1(a+b_{RCV}T_{SD})}{b_{RCV}}\left\{1+\frac{2b_{LOG}\omega\phi M_1(a+b_{CP}T_{SD})}{b_{CP}\left[\rho_{SD}^{M}-\omega\phi M_1(a+b_{LOG}T_{SD})\right]}\right\}$$

$$+\frac{2\omega\phi D_{UP}M_1^2(a+b_{CP}T_{SD})}{b_{CP}P\left\{\rho_{SD}^{M}-\omega\phi M_1(a+b_{LOG}T_{SD})\right\}} \tag{6.15}$$

式（6.14）、（6.15）はそれぞれＣＰＵに起因する条件とＳＤに起因する条件を示す。方式１ではリカバリ時間が与えられると両式を同時に満足する容量M_1のデータベースを収容できる。

T_{RCV1}とM_1の関係を図６．５に示す。図中の実線は式（6.14）、（6.15）の両方を満足するM_1を表し、破線は式（6.15）を満足するM_1を表している。式（6.14）を満足するM_1はほぼ実線と同一となるため省略したが、T_{RCV1}に比例してM_1が増加する領域では、わずかな差ではあるがＳＤによってM_1が決まる。M_1が飽和する領域ではＣＰＵによってM_1が決まる。これはM_1が飽和値に近づくと、ＣＰＤＢ取得のための負荷をＣＰＵに割り当てることが難しくなり、急激なT_{CP}の延長とＬＯＧ量の増加を引き起こすためである。

方式２で収容可能な最大データベース容量M_2は、式（6.6）、（6.13）より、次式となる。

$$M_2=\frac{\rho_{CPU}^{M}P}{\phi D_2} \tag{6.16}$$

方式１、２で収容可能な最大データベース容量の比をα（$=M_1/M_2$）とおく。αが１より大きい場合は方式１が有利であり、αが１より小さい場合は方式２が有利となる。ϕとαの関係を図６．６に示す。図６．６から判るように、要求されるリカバリ時間T_{RCV}を６０秒とすると、ϕが０．５（Tr/MB/秒）以上であれば方式１の方が、０．５（Tr/MB/秒）以下であれば方式２の方がより多くのデータベースを収容できる。

図６．７はϕとT_{RCV}から算出した方式１、２の適用領域を示す。図６．７より、デー

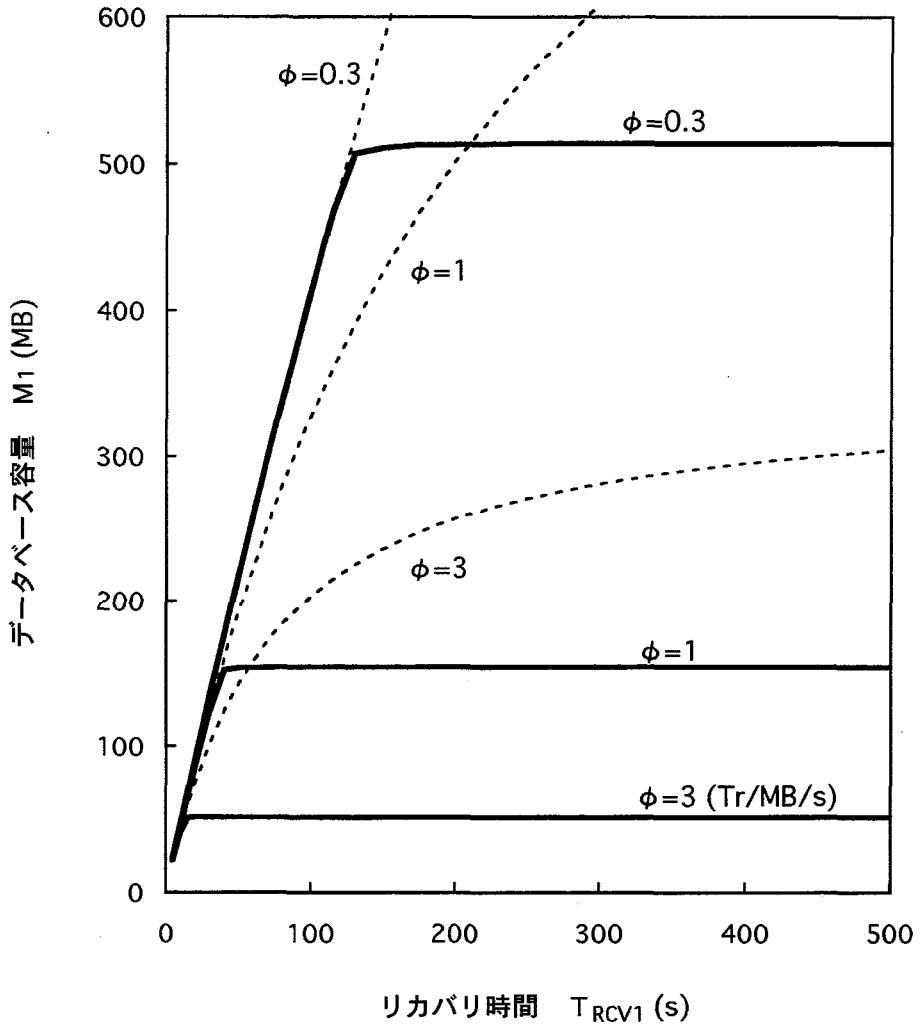


図6.5　リカバリ時間とデータベース容量

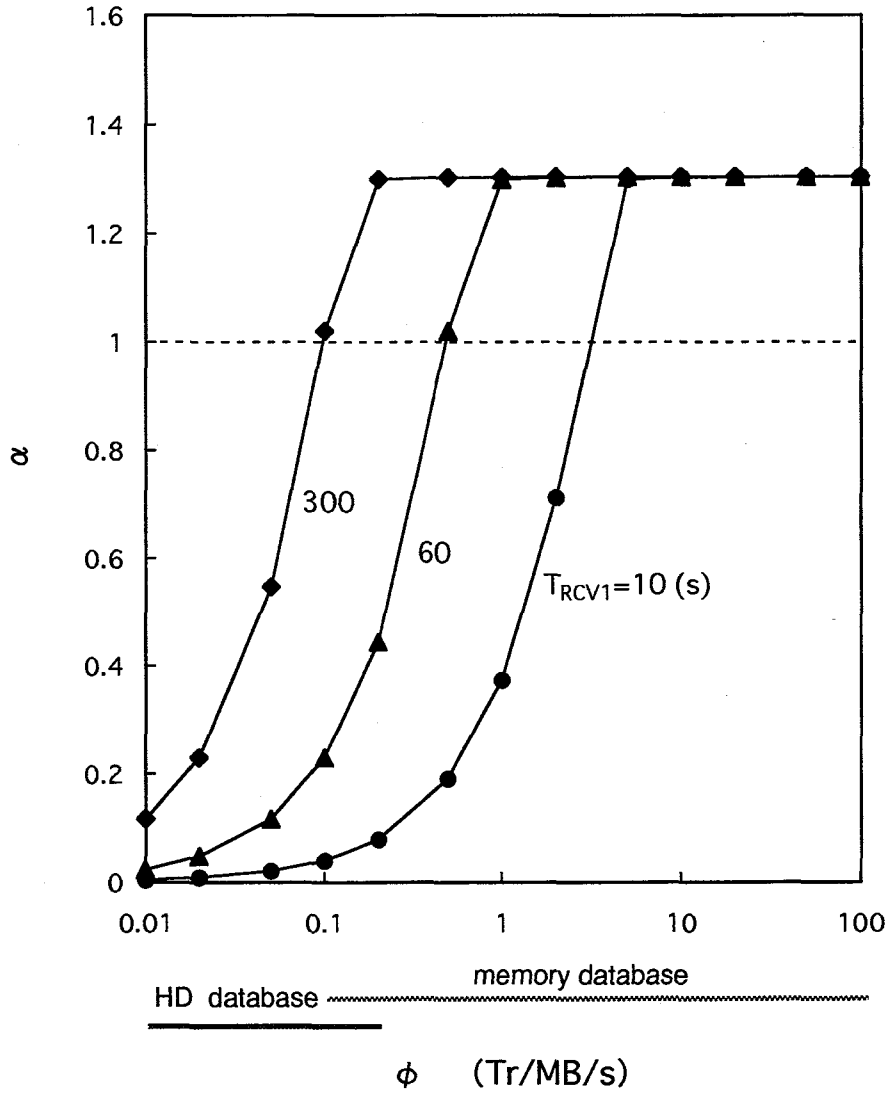

1.6
1.4
1.2
1
0.8
0.6
0.4
0.2
0
α
300
60
T_{RCV1}=10 (s)
0.01
0.1
1
10
100
HD database
memory database
φ (Tr/MB/s)

図6.6　φとαの関係

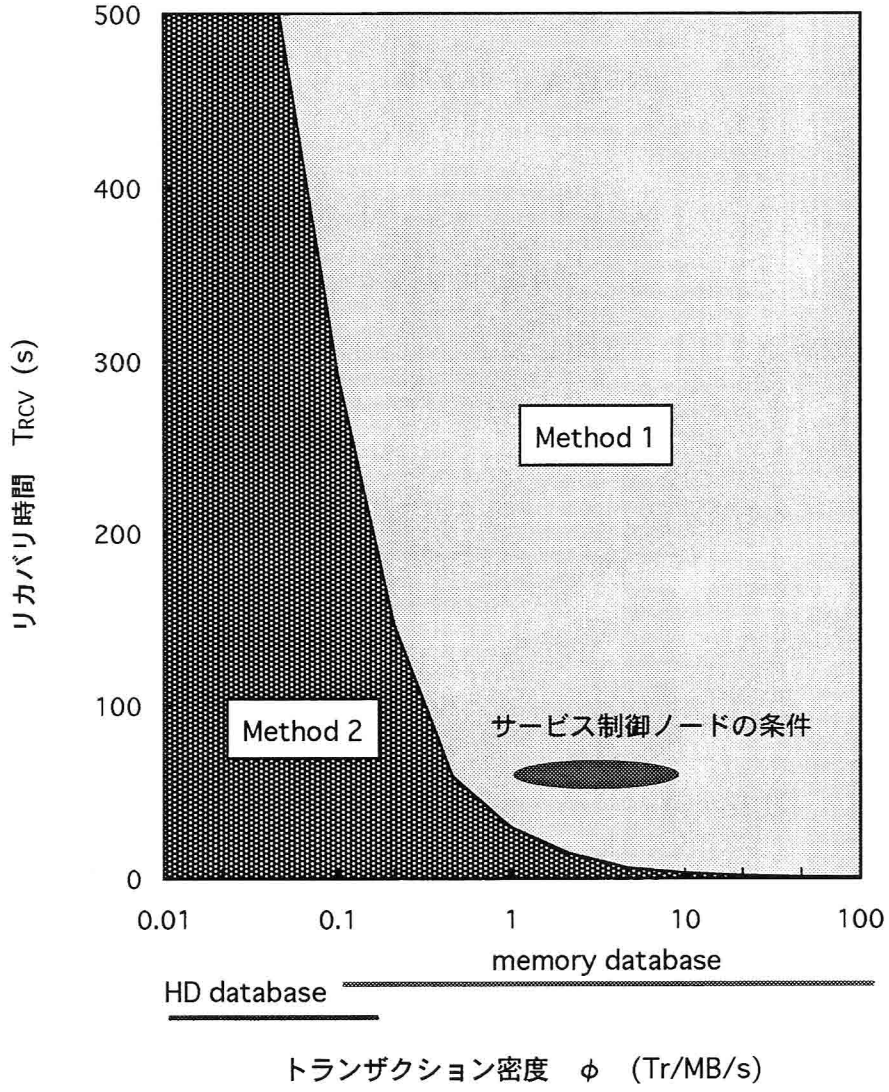


図６．７　リカバリ方式の適用領域

タベースのトランザクション密度が高く、許容されるリカバリ時間が長いほど方式１を適用する方が有利であり、逆の場合は方式２を適用する方が有利となる。ϕが０．１（Tr/MB/秒）以上でなければメモリデータベースの実現は難しいことから、メモリデータベースに適する領域では広範囲にわたり方式１が有利となる。

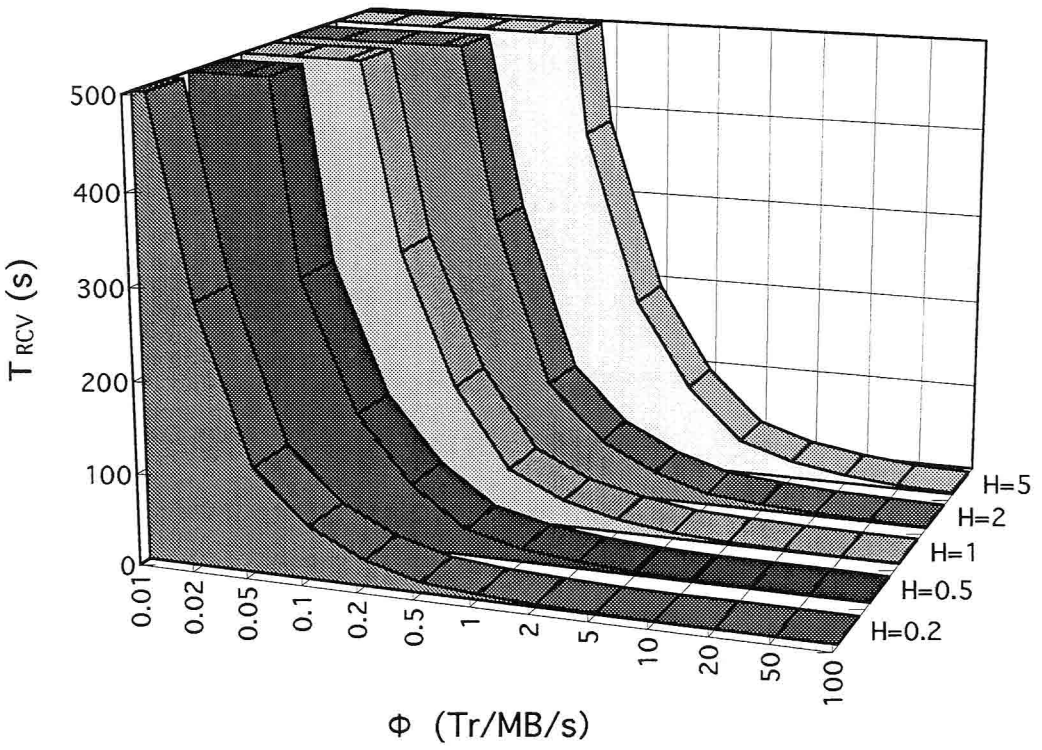
６．４．３．２　ハードウェア性能向上の影響

ＣＰＵ性能、ＳＤ性能は今後とも向上することが予想される。ＣＰＵ、ＳＤの性能向上の度合いをh_{CPU}、h_{SD}とし、将来、ＣＰＵ性能が$h_{CPU}\cdot P$、ＳＤ性能がa/h_{SD}、T_{SD}/h_{SD}となったとする。また、ＣＰＵ、ＳＤの性能向上度の比をＨ（$=h_{CPU}/h_{SD}$）とすると、式（6.14）、式（6.15）は以下の通りとなる。

$$T_{RCV1} \geq \frac{(\frac{M_1}{h_{CPU}})H(a+b_{RCV}T_{SD})}{b_{RCV}}\left\{1+\frac{2b_{LOG}\omega\phi D_{IO}(\frac{M_1}{h_{CPU}})}{b_{CP}\left[\rho_{CPU}^{M}P-\phi D_1(\frac{M_1}{h_{CPU}})\right]}\right\} + \frac{2\omega\phi D_{UP}D_{IO}(\frac{M_1}{h_{CPU}})^2}{b_{CP}P\left\{\rho_{CPU}^{M}P-\phi D_1(\frac{M_1}{h_{CPU}})\right\}} \tag{6.17}$$

$$T_{RCV1} \geq \frac{(\frac{M_1}{h_{CPU}})H(a+b_{RCV}T_{SD})}{b_{RCV}}\left\{1+\frac{2b_{LOG}\omega\phi(\frac{M_1}{h_{CPU}})H(a+b_{CP}T_{SD})}{b_{CP}\left[\rho_{SD}^{M}-\omega\phi(\frac{M_1}{h_{CPU}})H(a+b_{LOG}T_{SD})\right]}\right\} + \frac{2\omega\phi D_{UP}(\frac{M_1}{h_{CPU}})^2H(a+b_{CP}T_{SD})}{b_{CP}P\left\{\rho_{SD}^{M}-\omega\phi(\frac{M_1}{h_{CPU}})H(a+b_{LOG}T_{SD})\right\}} \tag{6.18}$$

図６．８はϕとT_{RCV}が与えられた場合、方式１と方式２の優劣がＨによってどのように変化するかを表している。将来、ＣＰＵ、ＳＤの性能が向上したとしても、その向



shadow zone　：$\alpha < 1$ (method 2)
no shadow zone：$\alpha > 1$ (method 1)

図６.８　リカバリ方式の適用領域
（ハード性能との関係）

上度度合いが同一、すなわちH＝１であれば、式（6.14）、（6.15）と式（6.17）、（6.18）はM_1が（M_1／h_{CPU}）となる以外は全く同一の式となる。これは、リカバリ時間を同一とした場合、M_1をh_{CPU}倍大きくできることを意味している。方式２ではh_{CPU}に比例してM_2が増加する。方式１、方式２共に収容可能なデータベース容量の向上度が同一となるため、現時点の評価結果がそのまま当てはまる。また、図６．８からわかるように、Hが大きくなるに従って方式２の適用領域が拡大し、逆に小さくなるに従って方式１の適用領域が拡大する。

６．５　コスト評価

方式１、２をコスト面から比較評価する。図６．１に示すように、モジュールを構成するプロセッサ、主メモリ、通信チャネル等は方式１、２に係わらず必要となる。このようにモジュールのコストを構成する主要部分は両方式で同様に必要とされる。方式１と方式２のモジュール構成上の主要な相違点は、方式１の場合はチェックポイントデータベース（ＣＰＤＢ）やデータベースの更新ログ（ＬＯＧ）を取得するため半導体ディスク装置が必要であること、方式２の場合はバックアップする相手モジュールのデータベースのコピー（ＤＢＣ）を主メモリに常駐させておくため主メモリの記憶容量が多く必要となることのみである。半導体ディスク装置と主メモリの記憶媒体の差や必要とされる記憶容量に若干の差があるとしても、モジュールのコストへの影響は少なく、方式１、２でモジュールコストはほとんど変わらない。

モジュールコストが同一であるとすれば、システムとしてのコストはモジュール台数に依存して決まる。モジュール台数はシステムに要求されるスループットを、方式１、２のモジュール１台当たりのスループットλ_1、λ_2で割った値となる。前節の性能評価で述べたように、スループットは方式１が方式２に比べ約３０％高いため、システムとしてのコストは方式１が方式２に比べ約３０％低くなる。

6.6　信頼性評価

6.6.1　不揮発化

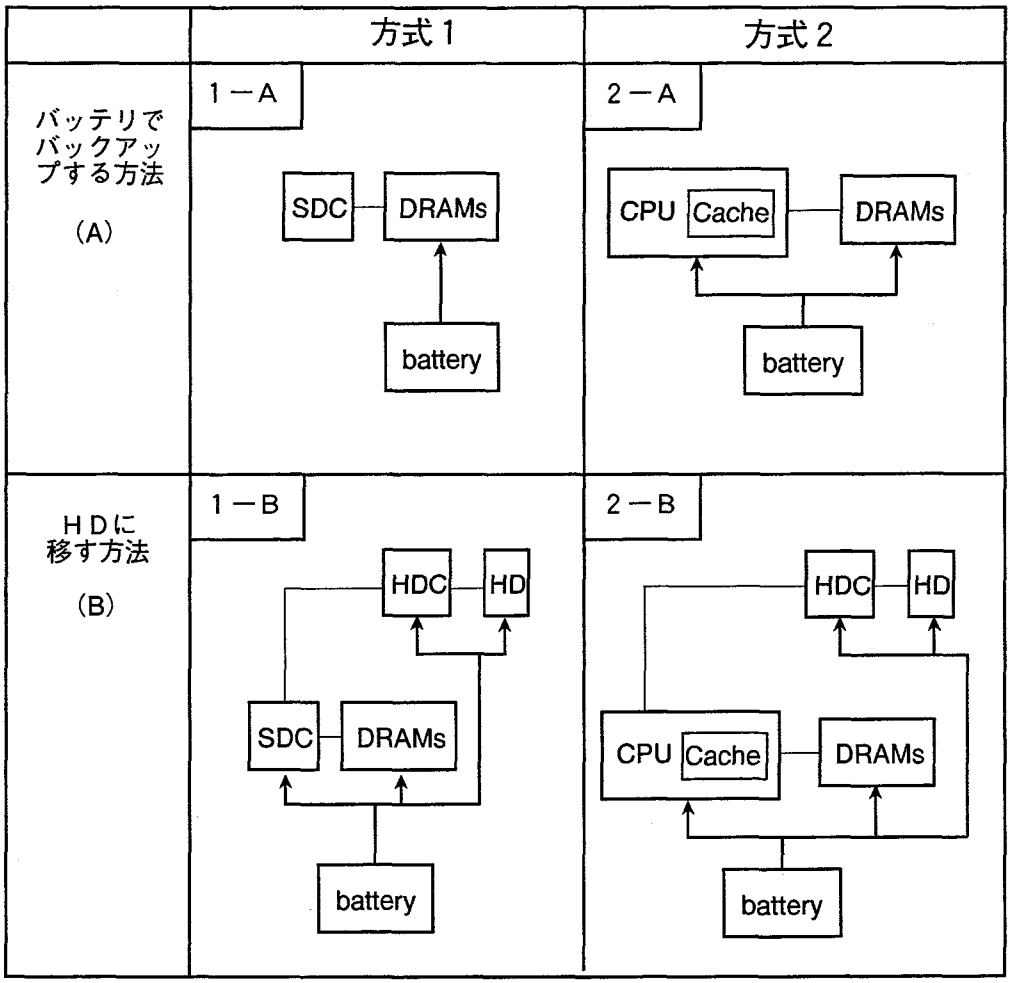
方式１、２ともにメモリ素子としてＤＲＡＭを使用するため、電源の供給が停止した場合には主メモリ上のデータベースはもとより、半導体ディスク装置または他のモジュールの主メモリに格納しているバックアップ情報も消失する。電源系の障害は、①モジュール内の電源部の障害、②モジュールへの配電系統の障害、③局舎全体の電源障害に分類できる。

方式１の場合、半導体ディスク装置および電源部を二重化し、配電系を別にすることにより①、②の部分の一重障害では情報は消失せず、情報の消失は二重障害のみとなる。方式２の場合、相互にバックアップの関係にあるモジュールの電源部は別であり、モジュールへの配電系を別にすることにより同様に①、②の一重障害では情報の消失を回避できる。しかし、両方式とも、このままでは③の障害には対応できない。局舎電源を構成する各装置についても、通常、多重化等により高信頼化がはかられているが、最近の１０年間においても、落雷や増設工事中の事故等により、局舎全体の電源断が発生している。電源断の時間は、ほとんどが１～２時間以内であり、極くまれではあるが、１０時間程度となった場合がある。情報の消失を防止するためには固有の電源による不揮発化が必要である。

不揮発化の方法としては、電源供給停止後、バッテリによりバックアップする方法（方法Ａ）とＤＲＡＭの内容を磁気ディスク装置（ＨＤ）に移す方法（方法Ｂ）がある。前者はバックアップ時間Ｔが長くなるに従って、バッテリ量が大きくなる問題があるが、装置構成は簡単である。一方、後者は一度ＨＤに格納すれば、バックアップ時間の制約は無い。しかし、データをＨＤに移す間はＨＤや書込み／読出しを制御するコントローラを含め、バッテリによるバックアップが必要となる。

方式１、２の不揮発化方法を表６.２に示す。方法２－Ａでは、最新の情報がキャッシュメモリ上にある場合があり、キャッシュメモリの内容を主メモリに掃き出した

表６．２　不揮発化方法


方式１
方式２
バッテリで
バックアッ
プする方法
(A)
１－A
SDC
DRAMs
battery
２－A
CPU
Cache
DRAMs
battery
ＨＤに
移す方法
(B)
１－B
HDC
HD
SDC
DRAMs
battery
２－B
HDC
HD
CPU
Cache
DRAMs
battery

あと、主メモリのＤＲＡＭのみを不揮発化する必要がある（付録６．３参照）。各方法の消費電力は次式で表せる。記号の内容を表６．３に示す。

$$E_{1-A} = p_{MS}BST/m \tag{6.19}$$

$$E_{1-B} = (p_{MA}BS/m + p_{SDC} + p_{HDC} + p_{HD})S/M_{HD} \tag{6.20}$$

$$E_{2-A} = (p_{MA}BS/m + p_{CPU})T_0 + p_{MS}BS(T-T_0)/m \tag{6.21}$$

$$E_{2-B} = (p_{MA}BS/m + p_{CPU} + p_{HDC} + p_{HD})S/M_{HD} \tag{6.22}$$

バックアップ時間と消費電力の関係を図６．９に示す。地震、火災等の特別な障害を除けば、局舎の電源が長時間に渡って障害になることは少なく、情報のバックアップ時間は２０時間程度とした。図６．９より、バックアップ時間がこの程度であればいずれの方法も消費電力の差はほとんど無い。方法ＢはバッテリのほかにＨＤおよびＨＤ制御装置が必要であり、ハードウェア量の点では方法Ａが有利である。これに加えて、ＨＤは可動部を有しており、信頼性、保守性の面でも方法Ｂは劣る。全体としていずれの方法もバッテリ量の差は小さいが、可動部を持たないこと、構成が最も簡単なことから方法１－Ａの実用性が高い。

要求される電力容量がこの程度まで小さくなると、一般に大量に使用されているニッカド電池を使用可能となる。ニッカド電池はシール鉛電池に比べて単位重量当たりの電力容量（重量電力密度）が大きく、装置に組み込むことが容易である。また、ニッカド電池の寿命（約５年程度）はシール鉛電池に比べ数年長い。

６．６．２　高信頼化構成

モジュールとしての耐障害性、冗長化構成の選択の２面から方式１、２について簡単に比較する。

６．６．２．１　モジュールとしての耐障害性

データベースのリカバリのために増加するメモリの容量は方式１の方がＬＯＧの

表６．３　　不揮発化評価パラメータ

記号	内容	設定値
P_{MA}	動作時のＤＲＡＭの消費電力	0.25w
P_{MS}	スタンバイ時のＤＲＡＭの消費電力	1.9×10^{-3}w
S	記憶容量	500MB
m	ＤＲＡＭの記憶容量	16Mbit
B	１バイトを実現するために必要なビット数	10bit/Byte
P_{SDC}	半導体ディスク装置の消費電力	40w
P_{CPU}	プロセッサの消費電力	100w
P_{HDC}	磁気ディスク制御装置の消費電力	40w
P_{HD}	磁気ディスク装置の消費電力	30w
M_{HD}	磁気ディスク装置のデータ転送速度	4MB/s
T_0	キャッシュメモリの内容を主メモリに掃き出す時間	1s

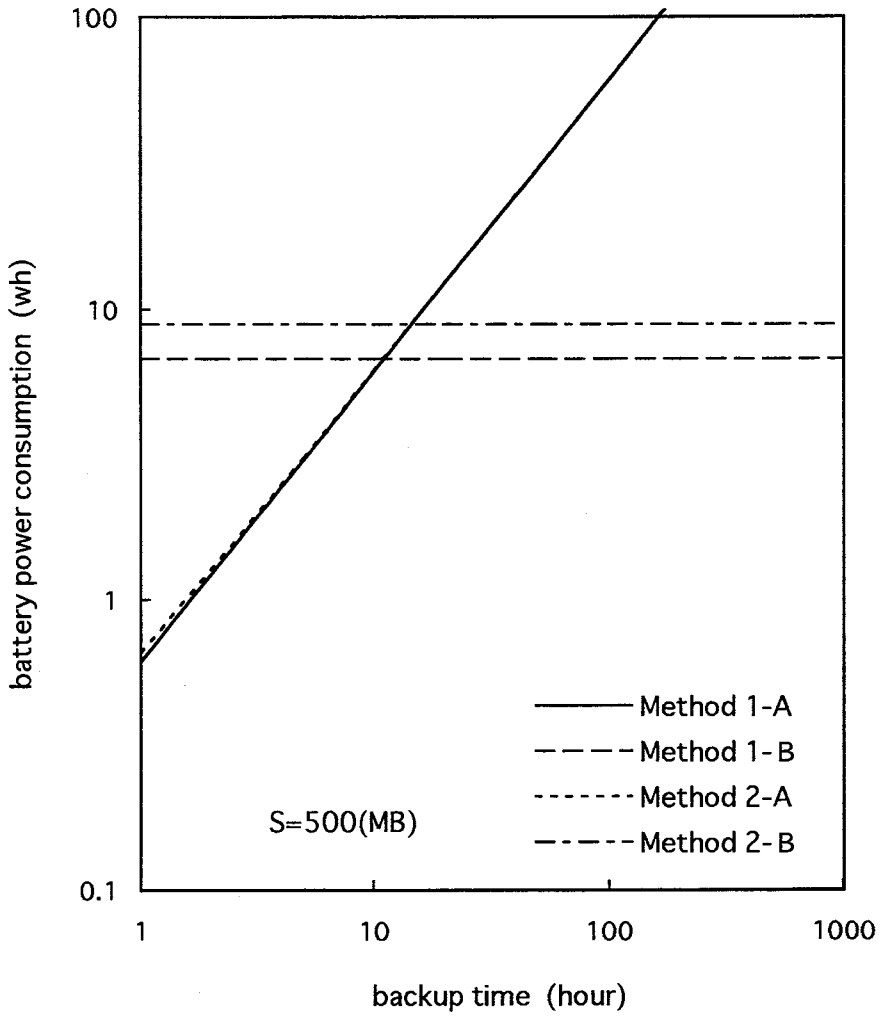


図6.9　バックアップ時間とバッテリの消費電力

容量分多くなるが、モジュール全体で見ると、プロセッサ系がＦＩＴ数の主要部を占めるため、方式１、２でＦＩＴ数の差は小さい。このため、ハードウェア障害に起因するモジュール障害発生の度合いはほとんど同じとなる。

ＳＣＰではシステム稼働中にファイル更新等の保守を行う必要がある。方式２ではトランザクション処理毎にバックアップ側のＤＢを最新の状態に更新してしまうため、稼働中のファイル更新の際のバグや操作ミスにより誤ってＤＢを破壊した場合には、ＤＢを元に戻すことはできない。このため、ＣＰＤＢを事前に取得する等の何らかの新しい処置が必要となる。これに対して、方式１ではＣＰＤＢとＬＯＧを残しており、ＤＢ破壊が発生しても、ＣＰＤＢやＬＯＧを用いてＤＢ破壊を限定し、復元することができる。また、ＳＤの容量を増加させることでT_{CP}とは独立に、ＣＰＤＢの内容を多重に必要な時期まで保存しておくことが容易である。このため、方式１の方がサービス中のソフト保守はしやすい。

６.６.２.２　冗長化構成とプロセッサ稼働率

ノード内バックアップ方法として、１対１の相互バックアップのほかに、１モジュールが障害となった場合、障害となったモジュールの負荷を他のｎモジュール（ｎ≧２）で分担する方法もある。プロセッサ稼働率はｎ／（ｎ＋１）となり、ｎの増加と共に高く設定できる。方式１ではＳＤを他のモジュールからアクセス可能とする必要があり、ｎに比例してＳＤのポート数が増加し、ハード構造が複雑になる。一方、方式２はモジュール間結合機構を介して他のモジュールと結合されており、ハード構成上、ｎの増加に対して柔軟に対応できる。しかし、１モジュールの障害を複数のモジュールがバックアップすると、障害発生時の切り替えや障害装置回復時の切り戻しのソフト制御が極めて複雑となる。このため、実用性の高いシステムとして構築するためには両方式共に１対１の相互バックアップとし、１／２のプロセッサ稼働率で動作させる方が望ましい。

加えて、地震、火災などの災害に対応するためには、地理的に離れた２つの地点に

設置されたノード間での相互バックアップが必要となる。この場合、どちらかのノードが障害となった時には、別のノードがすべての負荷を処理する必要があり、通常は１／２のプロセッサ稼働率で動作させる。第３章ではノード内バックアップとノード間バックアップのためのプロセッサ稼働率の余裕を共用させることにより、効率的なバックアップ方法を提案した。プロセッサの稼働率をノード間バックアップに合わせて１／２に設定し、ノード内バックアップ時にもこの１／２のプロセッサ稼働率の余裕を活用する方法である。ノード間バックアップとノード内バックアップが同時に発生する確率は極めて低いため、経済的にシステムの高信頼化を達成できる。ノード間バックアップとノード内バックアップの両方を組合せた冗長構成を採る場合は、ノード内バックアップ方法に依存せず、方式１、２に係わらずプロセッサ稼働率は同一となる。

６.７　実現方式

前節までの検討を踏まえ、高度ＩＮ用ＳＣＰとして試作したモジュールの方式と構成について述べる。

リカバリ方式としては半導体ディスク装置にＬＯＧとＣＰＤＢを取得する方式１を採用した。また、半導体ディスク装置の不揮発化はＤＲＡＭのみをバッテリによりバックアップする方法とした。これは主に次の理由による。

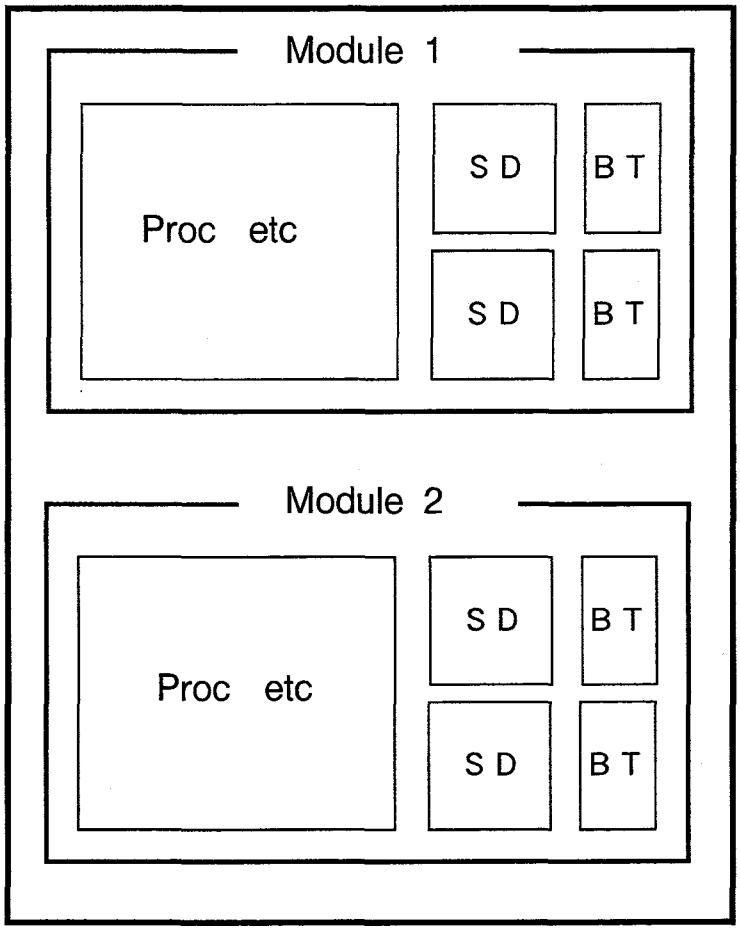
高度ＩＮのデータベースのトランザクション密度は１～１０（Tr/MB/秒）程度であり、要求されるリカバリ時間は６０秒程度である。この条件では、図６.７から判るように明らかに方式１の方がより多くのデータベースを収容できる。また、２０時間程度の不揮発化ではＳＤをバッテリによりバックアップする方法が最も構成が簡単でハードウェア量が少ない。高信頼化のための冗長化構成の面からは、高度ＩＮではノード間バックアップとノード内バックアップの併用が必要であり、この場合、方式１、２でプロセッサ稼働率の差はない。加えて、ソフト障害に対処する能力は方式１

の方が高い。

システムとしてのバックアップ動作の確認等を行うためモジュールを試作した。1つの架の中に2台のモジュールを搭載し、同一架内のモジュール間で、ＳＤを介して相互にバックアップ可能とした。バッテリはＣＰＵ、ＳＤと同一バックボードに搭載し、全体としてコンパクトな実装が可能となった。モジュールの実装構成を図６．１０に示す。試作を通して、電源断時に正常にバックアップに移行して２０時間のバックアップに耐え得ること、電池の周囲温度の上昇が室温に比べて１０℃以下に収まること、およびコンピュータ等に適用されているＶＣＣＩの伝導雑音や放射雑音の規定値を満足すること等を確認した。

ＣＰＵやＳＤの論理パッケージは故障検出回路を有しており、その保守は、故障が検出された時点での報告にもとづき予備パッケージと交換して正常性を確認することにより行われる。一方、バッテリには寿命があり、ある時間経過後には使用できなくなる。このため、通常は寿命が切れる前に定期的に交換する必要がある。モジュールを構成するＣＰＵ、ＳＤ等の保守とバッテリの保守が異なることは、2通りの保守体制をとる必要があり、保守が煩雑となる問題がある。加えて、バッテリの寿命は周囲の温度など使用状況に応じて数倍変動する。このため、定期保守する場合は、大半のバッテリではバックアップ能力がまだ残っているとしても、劣化が早く寿命が切れると予想されるごく少数のバッテリに合わせて、交換周期を短く設定する必要がある。

これらを解決するため、バッテリの劣化判定機構を開発し[63]、ＣＰＵ、ＳＤ等の論理回路から構成されるパッケージと同様に障害や劣化の報告を受けてから交換可能とするとともに、平均的にバッテリを長く使用できるようにした。ここで、劣化判定とは、バッテリが所望の時間バックアップに耐えられるか否かを判定し、耐えられないと判断した場合、交換が必要なことを遠隔の保守センタに報告する機能である。図６．１１に劣化判定機構を組込んだバッテリの構成を示す。バッテリは単３のニッカド電池６個を１パックとし、４パック（電池総数は２４個）を使用して、論理回路を搭載するパッケージ（３０×３３ｃｍ^2）と同一のものに実装し、パッケージ２枚分の体積で実現できた。劣化判定は１パック毎に図６．１１のスイッチをテスト側（T



Proc: Processor
SD: Semiconductor disk device
BT: Battery

図6.10　モジュールの実装構成

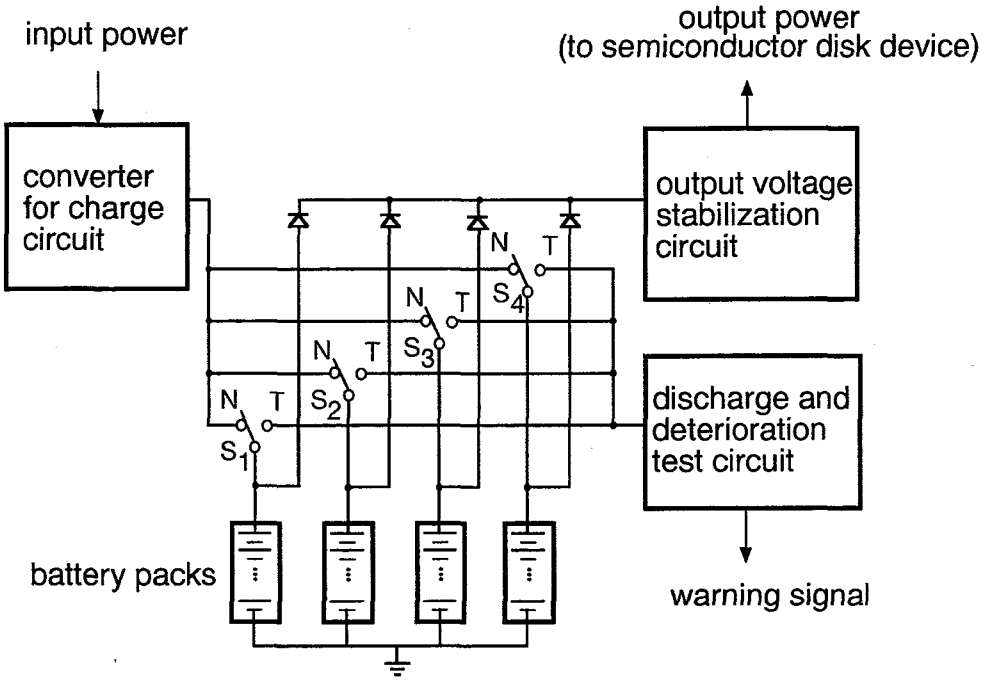


図６．１１　バッテリの構成

側）にセットし放電回路を通して放電させ、経過時間と電圧低下からバッテリのバックアップ能力が２０時間以下となったとき劣化と判断する。１パックの判定が終了して正常であれば、スイッチを充電側（Ｎ側）に倒して充電する。充電が終了した時点で次のパックの劣化試験を行う。これにより劣化試験中に電源障害が発生した場合でも残りの３パックで半導体ディスク装置のバックアップを可能とした。１パックの劣化判定と再充電に約３日を要し、約半月で４パックの試験が終了する。この試験を３ヵ月周期で行うこととした。

６．８　結言

メモリデータベースのリカバリのため、トランザクション毎のＬＯＧを取得する記憶装置として、半導体ディスク装置を適用する方式と他のモジュールの主メモリを適用する方式を代替案として設定し、性能、信頼性の２つの面から総合的に比較評価を行った。性能に関しては、スループットとリカバリ時間を併せて、モジュール毎に収容可能なデータベース容量を評価し、データベースのトランザクション密度と要求されるリカバリ時間が与えられた場合の適用領域を明らかにした。信頼性に関しては、不揮発化方法および冗長化構成法について両方式の比較評価を行った。また、バッテリバックアップについては劣化判定機構を新たに開発し、論理パッケージと同様に障害や劣化の報告を受けてから保守可能とし、バッテリ導入上の問題を軽減する方法を提案した。その結果、高度ＩＮのメモリデータベースのリカバリ方式としてバッテリバックアップを行った半導体ディスク装置による実用性の高い方式を明らかにした。なお、本章では、ＬＯＧはリカバリに必要な時点を過ぎれば破棄するとして検討を進めたが、ＯＬＴＰシステムの中には数時間前の状態からリカバリを可能とするためや、リカバリ以外の用途に使用するため、ＬＯＧを長時間に渡って保持する必要のあるアプリケーションもある。この場合は、本章のモデルに、ＬＯＧを大容量の記憶媒体に取得する処理を追加することにより容易に拡張できる。

第７章
相互にバックアップされたサービス制御ノードの高信頼化運転保守技術

７．１　緒言

高い信頼性が要求される通信システムや、バンキングシステム、クレジット照会システム等のオンライントランザクション処理（ＯＬＴＰ）システムでは、従来からノード内で二重化する冗長構成が一般にとられている[21][55]。しかし、データベースを用いて全国を対象にサービスを行うようなシステムでは、ノード内の二重化のみでは、地震、火災等の大規模災害に対しては十分でなく、地理的に離れた２つの地点にノードを設置して、ノード相互でデータベース上の情報を含めてバックアップする冗長構成がとられるようになってきている[23][24][35]。インテリジェントネットワークの急激な普及や、ＯＬＴＰシステムでのサービス内容の多様化、高度化に伴って、このような形態のシステムが増加する傾向にある。

一方、システムの信頼性は、このような高信頼化のための冗長方式とならんで保守方式により大きく変動する。ノードに保守チームが常駐している場合は、障害検出と同時に修理に取り掛かることが可能である。しかし、ノード毎に保守チームを常駐させると保守コストの増大を招くため、通常、保守センタに保守チームを集中し、複数のノードを少数のチームで保守する方法が採られている。この場合、サービス停止につながるような障害では常駐の保守チームにより緊急に駆付けて対応し、サービス停止に至らない軽度の障害ではある程度時間的な余裕を持って平日勤務の保守チームにより対応する等、システムの緊急度合いに応じて駆付け時間を調整することが多い。

地理的に離れたノード間での相互バックアップは大規模災害への対応が本来の目的であるが、通常の装置障害に対しては、冗長度が上がるため、高い信頼性が容易に達成可能となる。許容される範囲での信頼性低下のもとで、駆付け時間の長時間化を図ることにより、保守の効率化とシステムの信頼性をバランス良く実現することが期

待できる。

本章では、データベースを持ち二重化された２つのノードを離れた地点に配置し、相互にバックアップするシステムを対象として、駆付け保守の方法と信頼性の関係を検討し、保守の効率化方法について考察する[64][65]。はじめに、ノード内で二重化された２つのノードが相互にバックアップするシステムを駆付けにより保守する場合のシステムの構成と信頼性算出方法を示す。既に、渡辺らはノード内で二重化した交換機と宅内装置からなる加入者系を対象として、駆付け保守による修理遅延の影響について検討を行っている[37]。本章では、渡辺らの考え方を、二重化された２つのノードが相互にバックアップするような構成をとるシステムに拡張して用いている。次に、信頼度の時間的な変化に注目し、システムが障害となった場合、どの程度急いで駆付けるかをシステム内の障害装置数や保守チームが到着しているか否かの状態によりレベル分けすることを提案し、どのレベルまで緊急駆付けを行うかに応じて、信頼性や駆付け回数の変化を明らかにする。これらの結果に基づいて、システムに要求される信頼度を満足した上で、非緊急駆付けについては複数の障害をスケジュール化可能で、直ちに駆付けなければならない緊急駆付け回数の少ない効率的な保守方法を明らかにする。

７.２　評価対象システムの構成と動作

７.２.１　評価対象システムの信頼度構成

本章では、データベースを保有し、トランザクション処理や呼処理を行うモジュールを基本の構成単位とする。ここで、モジュールは本章での障害検出、修理の基本の構成単位であり、プロセッサ、メモリ、ファイル装置、通信装置等、一式から構成される。ノード内ではモジュールが二重化され、さらに地理的に離れた２つのノードで１対１に相互にバックアップされる構成を信頼性評価の単位とする。以下、これをシ

ステムと記す。このような冗長構成がとられる例として、高い信頼性が要求される高度インテリジェントネットワーク（高度ＩＮ）やＯＬＴＰがある。高度ＩＮでは、電話機から発信された呼は交換機、共通線信号網を介してデータベースを保有するサービス制御ノード（ＳＣＰ）に集められる。また、ＯＬＴＰでは、端末からのトランザクションが通信回線を通じて処理センタに集められる等、ノードの設置に際して地理的な制約が少ないという特徴がある。

高度ＩＮの場合、全国的な大規模サービスを想定すると、ネットワーク内に数１０から数１００程度のシステムを配置する必要がある。ＯＬＴＰの場合は、通常、サービス毎には１システムから数システムで構成されるが、あるベンダが複数の顧客のシステムの保守を請負う場合には、多数のシステムの保守をまとめて行うこととなる。従って、保守対象は数１０から数１００の規模で配置されるシステム群とする。図７.１に評価対象システム全体の構成を示す。

高度ＩＮやＯＬＴＰのデータベースは、呼毎、トランザクション毎に更新されるデータベースであることから、異なる地点に配置されたノード間で相互にバックアップを行う場合は、データベースそのものを相互に持ち合い更新内容をバックアップ側のデータベースに反映する必要がある[15] [43]。本章では図７.１に示すように、２つのノードの対応するモジュール＃１-１-１と＃１-２-１および＃１-１-２と＃１-２-２がデータベースを相互に持ち合い、更新内容を報告し合うことによりバックアップ側のデータベースに反映するとした。

高度ＩＮでのパーソナル通信サービスのように単一サービスを多数のノードで分散して実現する必要がある場合のノード間バックアップ方法としては、１対１の相互バックアップのほかに、１ノードがダウンした場合、その負荷を複数のノードで分担する方法もある。この場合、ダウンしたノードの負荷が複数のノードに分散されるため、平均的にノードの稼働率を高く設定できるという長所がある。しかし、１ノードの障害を複数のノードがバックアップすると、通常動作時において、データベースの更新情報を常に複数のバックアップ側ノードに送信しなければならないことに加えて、ノードダウン発生時の切り替えや障害ノード回復時の切り戻しのソフトウェア制

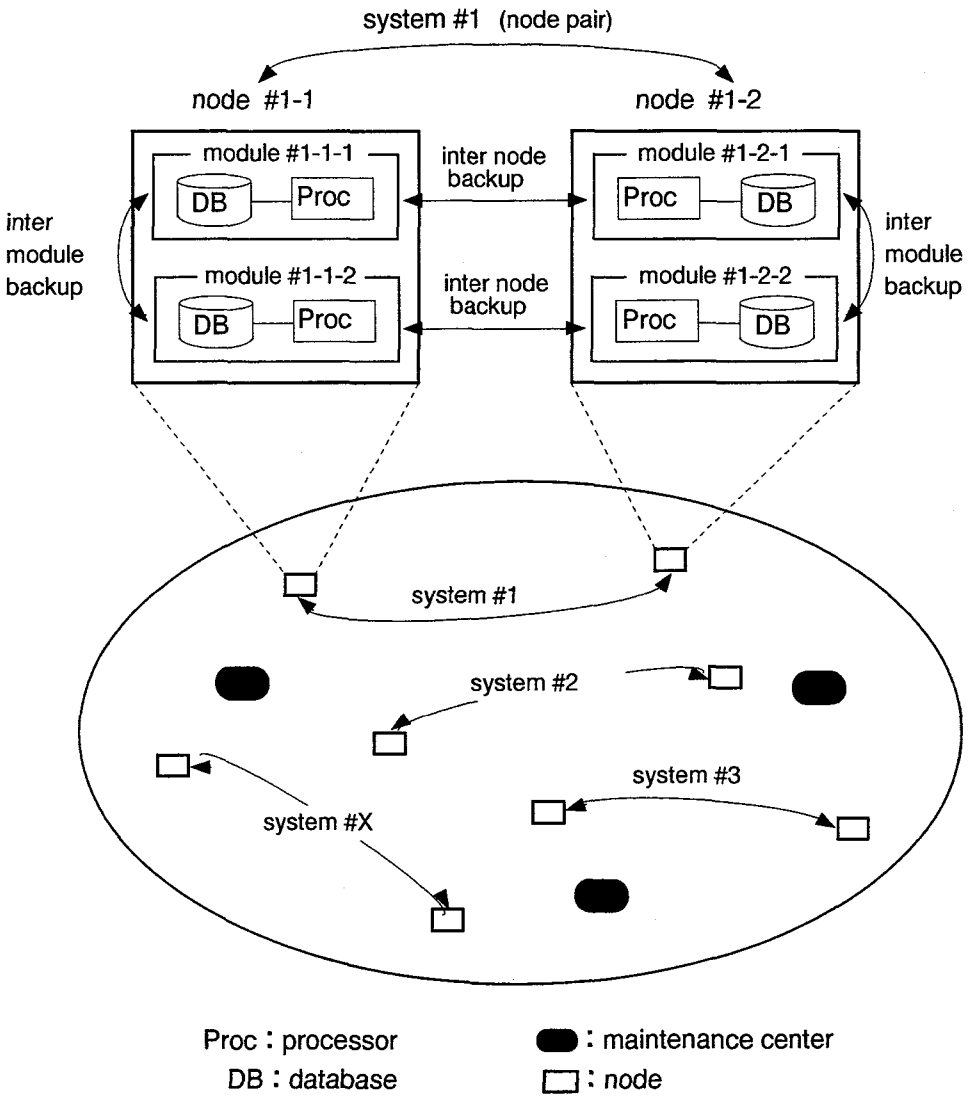


図7.1　評価対象システムの構成

御が複数のノードでほぼ同時に行われ、システム群の制御が極めて複雑となる。実用性の高いシステムとして、本章では1対1の相互バックアップとした。システムの信頼度構成を図7.2に示す。

7.2.2　モジュールの動作方法

一般に、冗長化により信頼度は向上するが、モジュールの稼働率が低下し、コストアップを招くこととなる。本章では、ノード単体での2重化構成の場合と同様のモジュール稼働率を実現するため、通常時はシステム内の4台（2台×2ノード）のモジュールをすべて動作させ、それぞれのノードで負荷を1／2ずつ分担するとした[(5)]。1台のモジュールが障害となった場合は、同一ノード内の別のモジュールが障害となったモジュールのデータベースを格納して負荷を処理する。同一ノード内の2台のモジュールが共に障害となった場合は、システム内のもう一方のノードのモジュールが共に正常であれば障害となったそれぞれのモジュールのデータベースを格納して負荷を処理する。もう一方のノードで1台のモジュールが障害となっている場合、すなわち、同一システム内で3台のモジュールが障害となった場合は、正常な1台のモジュールが2ノード分の負荷に対応し、すべてのデータベースを格納して処理する。この場合、正常な1台のモジュールに最大処理能力の2倍の負荷がかかるが、トラヒックの規制により両ノードにかかる負荷の1／2を処理するデグレード運転を行う。1台のモジュールがすべてのデータベースを保持することとなるが、このためのファイル、メモリ等の記憶容量は用意されているものとする。システム内の4台のモジュールがすべて障害となった場合はシステムダウンとなり当該システムで行っていたサービスが停止する。

モジュールの動作方法として、各モジュールが1／2の負荷を分担する方法のほかに、ノード内の2台のモジュールのうち1台のモジュールが1の負荷を分担し、別の1台のモジュールは待機させておく方法もある。この方法では、すべてのモジュールが正常であったとしても、負荷を分担したモジュールには、常に1の負荷がかかって

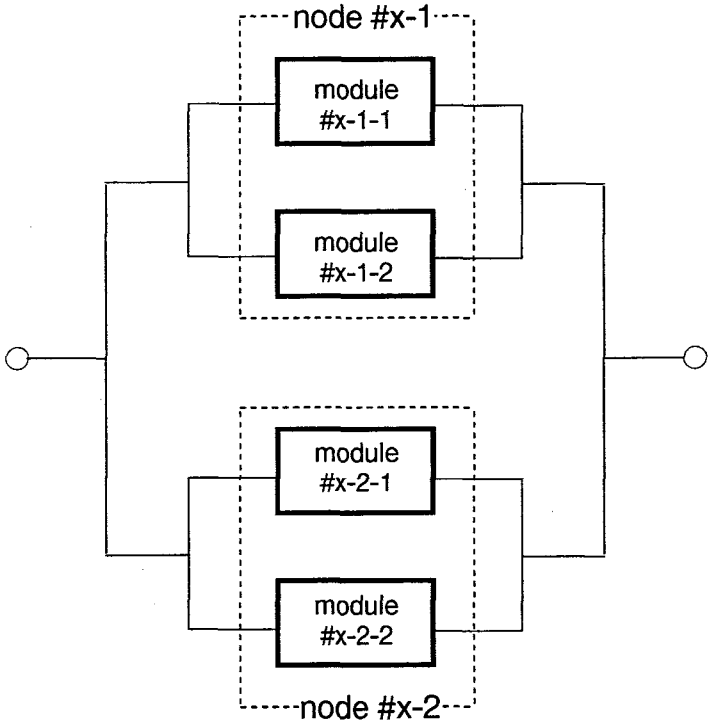


図7.2　システムの信頼度構成

おり、発生頻度は小さいとしても、突発的な負荷の増加に対してレスポンスタイムを保証することが難しい。一方、本章で想定したように、正常時には各モジュールが1／2の負荷を分担する方法では、モジュールの稼働率に余裕があるため、突発的な負荷の増加に対しても、レスポンスタイムの保証が十分可能である。

7.3　保守方法とシステムの状態

7.3.1　保守方法

保守チームは少数の保守センタに集中配置されるとする。各ノードは、その修理を担当するいずれか1つの保守センタに所属する。各保守センタは障害が報告（以下、駆付け要求と記す）されると保守チームが当該ノードへ駆付けて修理を行う。ソフトウェアの保守は遠隔から行われるものとし、本章ではハードウェアの保守のみを対象とする。渡辺らは二重化されたモジュールを対象とし、ノード内のモジュールが正常か修理中かの状態のほかに、保守チームが駆付けているか否かの状態を加えた状態遷移を提案している[37]。本章においても、保守チームの駆付けをモデルに組込むため、ノードの状態として保守チームがノードに駆付けているか否かを加えている。

当該ノードに保守チームがいない状態で障害が発生すると、保守センタへ駆付け要求が出される。当該ノードに保守チームが既にいる時は、新たな障害が発生しても、保守チームへの駆付け要求を出さず、既にいる保守チームが新たな障害を含めて修理を行うとした。但し、保守センタは担当するノードを含むシステム内の2ノードの状態を把握しており、駆付けの際、どの程度急ぐかは、次節以下に示すように2ノードの状態により判断可能とする。

修理は保守チームが到着した時点で開始する。一度あるノードに保守チームが駆付けた場合、当該ノードでの障害モジュール数に係わらず、すべてのモジュールを同時に修理できるものとする。すなわち、ノード内の2台のモジュールとも障害となった

場合、両方のモジュールを同時に修理する。保守チームはそのノードでの修理がすべて終了するまで，そのノードに留まって修理を行う。ノード内のすべての障害モジュールの修理が完了した場合は直ちに立ち去る。

7.3.2　システムの状態

システムの状態は各ノードでの障害モジュール数と保守チームが駆付けているか否かにより定義され$S_{ij,kl}$で表す。ここで、ｉ、ｋは各ノードの障害モジュール数、ｊ、ｌは各ノードに保守チームが駆付けているか否か（０の場合は駆付けていない、１の場合は駆付けている）を表す。２つのノードに対して同一の方法で保守を行うとすると$S_{ij,kl}$ の対称性から図７.３に示すようにシステムは１５通りの状態で表すことができる。具体的には、２つの状態$S_{i_1j_1,k_1l_1}$と$S_{i_2j_2,k_2l_2}$で$i_1=k_2$、$j_1=l_2$、$k_1=i_2$、$l_1=j_2$ が成立する場合は、２つの状態を$S_{i_1j_1,k_1l_1}$で縮退して表している。この縮退時の添字の表記としてはij,klを４桁の数字と見たときの小さい方を採用している。図７.３の横軸はシステム内の障害モジュール数（＝ｉ＋ｋ）、縦軸はシステムに駆付けている保守チーム数（＝ｊ＋ｌ）である。

モジュールの平均故障率および平均修理率（駆付けてから修理に要する平均時間の逆数）はシステムの状態によらず一定とするが、平均駆付け時間はシステムの状態毎に設定する。平均故障率をλ、平均修理率をμ、単位時間に保守チームの駆付ける率を$\omega_{ij,kl}$、すなわち、平均駆付け時間を$\omega_{ij,kl}^{-1}$ と表す。いずれも指数分布とする。各状態間の遷移は図７.３に示す通りとなる。状態$S_{ij,kl}$ の時刻ｔにおける確率を$P_{ij,kl}(t)$、その導関数を$P_{ij,kl}'(t)$ で表すと状態遷移の方程式は以下の通りとなる。

$$P_{00,00}'(t)=-4\lambda P_{00,00}(t)+\mu P_{00,11}(t) \tag{7.1}$$

$$P_{00,10}'(t)=4\lambda P_{00,00}(t)-(3\lambda+\omega_{00,10})P_{00,10}(t)+\mu P_{10,11}(t) \tag{7.2}$$

$$P_{00,11}'(t)=\omega_{00,10}P_{00,10}(t)-(3\lambda+\mu)P_{00,11}(t)+2\mu P_{00,21}(t)+2\mu P_{11,11}(t) \tag{7.3}$$

$$P_{00,20}'(t)=\lambda P_{00,10}(t)-(2\lambda+\omega_{00,20})P_{00,20}(t)+\mu P_{11,20}(t) \tag{7.4}$$

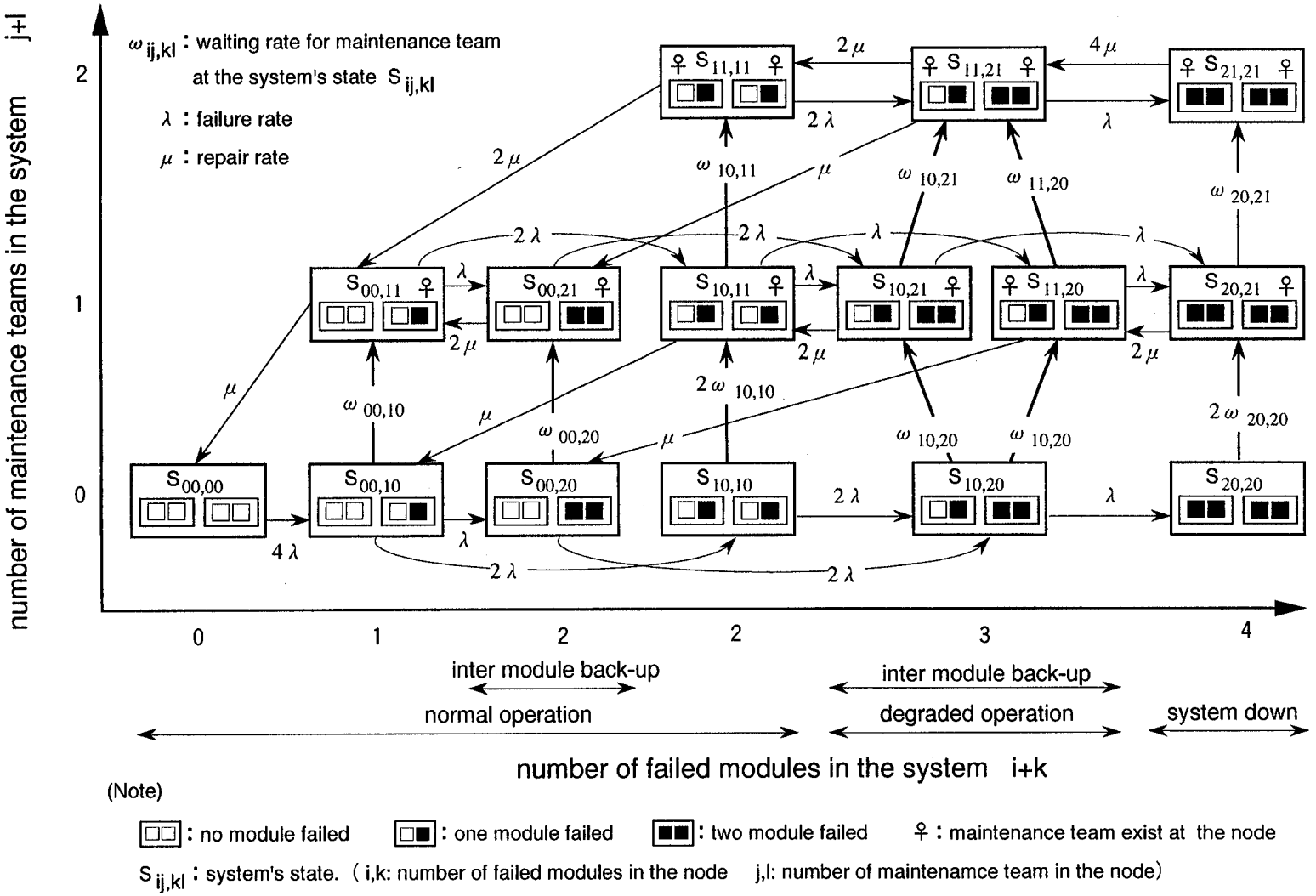

ω ij,kl : waiting rate for maintenance team at the system's state S ij,kl
λ : failure rate
μ : repair rate
number of maintenance teams in the system j+l
number of failed modules in the system i+k
inter module back-up
normal operation
inter module back-up
degraded operation
system down
(Note)
: no module failed
: one module failed
: two module failed
: maintenance team exist at the node
S ij,kl : system's state. (i,k: number of failed modules in the node j,l: number of maintenamce team in the node)

図７．３　状態遷移図

$$P_{00,21}'(t) = \lambda P_{00,11}(t) + \omega_{00,20} P_{00,20}(t) - (2\lambda + 2\mu) P_{00,21}(t) + \mu P_{11,21}(t) \tag{7.5}$$

$$P_{10,10}'(t) = 2\lambda P_{00,10}(t) - (2\lambda + 2\omega_{10,10}) P_{10,10}(t) \tag{7.6}$$

$$P_{10,11}'(t) = 2\lambda P_{00,11} + 2\omega_{10,10} P_{10,10}(t) - (2\lambda + \mu + \omega_{10,11}) P_{10,11}(t) + 2\mu P_{10,21}(t) \tag{7.7}$$

$$P_{11,11}'(t) = \omega_{10,11} P_{10,11}(t) - (2\lambda + 2\mu) P_{11,11}(t) + 2\mu P_{11,21}(t) \tag{7.8}$$

$$P_{10,20}'(t) = 2\lambda P_{00,20}(t) + 2\lambda P_{10,10}(t) - (\lambda + 2\omega_{10,20}) P_{10,20}(t) \tag{7.9}$$

$$P_{10,21}'(t) = 2\lambda P_{00,21}(t) + \lambda P_{10,11}(t) + \omega_{10,20} P_{10,20}(t) - (\lambda + 2\mu + \omega_{10,21}) P_{10,21}(t) \tag{7.10}$$

$$P_{11,20}'(t) = \lambda P_{10,11}(t) + \omega_{10,20} P_{10,20}(t) - (\lambda + \mu + \omega_{11,20}) P_{11,20}(t) + 2\mu P_{20,21}(t) \tag{7.11}$$

$$P_{11,21}'(t) = \omega_{10,21} P_{10,21}(t) + 2\lambda P_{11,11}(t) + \omega_{11,20} P_{11,20}(t) - (\lambda + 3\mu) P_{11,21}(t) + 4\mu P_{21,21}(t) \tag{7.12}$$

$$P_{20,20}'(t) = \lambda P_{10,20}(t) - 2\omega_{20,20} P_{20,20}(t) \tag{7.13}$$

$$P_{20,21}'(t) = \lambda P_{10,21}(t) + \lambda P_{11,20}(t) + 2\omega_{20,20} P_{20,20}(t) - (2\mu + \omega_{20,21}) P_{20,21}(t) \tag{7.14}$$

$$P_{21,21}'(t) = \lambda P_{11,21}(t) + \omega_{20,21} P_{20,21}(t) - 4\mu P_{21,21}(t) \tag{7.15}$$

$$\begin{aligned} &P_{00,00}(t) + P_{00,10}(t) + P_{00,11}(t) + P_{00,20}(t) + P_{00,21}(t) + P_{10,10}(t) + P_{10,11}(t) + P_{11,11}(t) \\ &+ P_{10,20}(t) + P_{10,21}(t) + P_{11,20}(t) + P_{11,21}(t) + P_{20,20}(t) + P_{20,21}(t) + P_{21,21}(t) = 1 \end{aligned} \tag{7.16}$$

7.4 信頼性の評価尺度と保守パラメータの設定

7.4.1 信頼性の評価尺度

サービスへ支障をきたしている状態として、4台のモジュールが障害となりサービスが停止している状態（以下、システムダウンと記す）と3台のモジュールが障害となりシステム全体の1／2の負荷を処理している状態（以下、デグレードと記す）の2通りがある。また、一方のノードがダウンしてノード間バックアップ状態となっている場合は、もう一方のノードで大規模災害が発生すると本来の目的が達成できない。また、大規模災害が生じないまでも、呼やトランザクションのノード間での転送、ノード回復後の切り戻し等、システムの運用が煩雑となるため、これらの状態について

も考慮する必要がある。各状態の発生確率と当該状態の継続時間の２つの面から、保守方式に対する信頼性の評価尺度として以下のものを用いる。

（１）**平均システムダウン時間と不稼働率**：システムダウンの開始から終了までの時間（システムダウン時間）の平均値と、サービス時間に対するシステムダウン時間の総計の占める割合

（２）**平均デグレード時間とデグレード率**：デグレードの開始から終了までの時間（デグレード時間）の平均値と、サービス時間に対するデグレード時間の総計の占める割合

（３）**平均ノード間バックアップ時間とノード間バックアップ率**：ノード間バックアップにより、１つのノードでのみサービスが行われている時間（ノード間バックアップ時間）の平均値と、サービス時間に対するノード間バックアップ時間の総計の占める割合

システムとしての不稼働率をP_{SDN}、平均システムダウン時間をT_{SDN}、デグレード率をP_{DEG}、平均デグレード時間をT_{DEG}、ノード間バックアップ率をP_{BUP}、平均ノード間バックアップ時間をT_{BUP}とすると、以下のように表される。ここで、定常状態、すなわち$P_{ij,kl}'(t)$が０のときの状態$S_{ij,kl}$の確率を$P_{ij,kl}$と表す。

$$P_{SDN} = P_{20,20} + P_{20,21} + P_{21,21} \tag{7.17}$$

$$T_{SDN} = P_{SDN} / \{\lambda(P_{10,20} + P_{10,21} + P_{11,20} + P_{11,21})\} \tag{7.18}$$

$$P_{DEG} = P_{10,20} + P_{10,21} + P_{11,20} + P_{11,21} \tag{7.19}$$

$$T_{DEG} = P_{DEG} / \{2\lambda(P_{00,20} + P_{00,21} + P_{10,10} + P_{10,11} + P_{11,11}) + 2\mu P_{20,21} + 4\mu P_{21,21}\} \tag{7.20}$$

$$P_{BUP} = P_{00,20} + P_{00,21} + P_{10,20} + P_{10,21} + P_{11,20} + P_{11,21} \tag{7.21}$$

$$T_{BUP} = P_{BUP} / \{\lambda(P_{00,10} + P_{00,11}) + 2\lambda(P_{10,10} + P_{10,11} + P_{11,11}) + 2\mu P_{20,21} + 4\mu P_{21,21}\} \tag{7.22}$$

7.4.2　信頼性目標と保守パラメータの設定

Ｇｒａｙらは、不稼働率の大小に基づいて、高信頼システムの簡単なクラス分けを行っている。たとえば、不稼働率が10^{-A}のシステムは、クラスＡの信頼度とし、現状の通信システムの信頼度はクラス５程度、すなわち不稼働率が10^{-5}程度としている。今後、より高信頼のシステムが要求され、その信頼度目標としてクラス７程度、すなわち不稼働率が10^{-7}程度とすることを提案している[43]。また、高度ＩＮの場合は、交換機の上位にサービス制御ノード（ＳＣＰ）を配置してサービスが行われるため、交換機とＳＣＰのそれぞれの不稼働率が加算されてサービスに影響する。高度ＩＮサービスを一般の電話サービスと同程度の信頼度でユーザに提供するためには、ＳＣＰの不稼働率は交換機の不稼働率（$\fallingdotseq 6\times10^{-6}$）[55]に比べて１～２桁小さく、無視できる程度であることが望ましい。これらの状況を踏まえて、二重化された２つのノードが相互にバックアップするシステムの不稼働率の目標は10^{-7}以下とする。

能條らは不稼働率を極めて小さくしたとしても同時に平均システムダウン時間に制限を加えることを提案している[57]。これは、システムダウンの発生がどんなにまれであるとしても、一度システムダウンとなった場合、それが長時間となることはサービスへの影響が大きくなるためである。ここでの提案も踏まえ、平均システムダウン時間については１～３時間程度とする。

デグレード状態とは、サービス停止には至っていないが、処理能力が１／２に低下している状態であり、極力発生頻度を押さえる方が望ましい。本章ではデグレード率は不稼働率に比べて３桁程度ゆるやかとし10^{-4}以下とする。これは１年間に１時間程度のデグレードに相当する。また、平均デグレード時間については、平均システムダウン時間の数倍を想定し、１０時間程度とする。

同一システム内の１つのノードが障害となり、残りの正常な１つのノードがサービスを行っているとき、サービス中の正常なノードが大規模災害で破壊されるとサービスが停止し、最悪、データベースが破壊される。この場合は、バックアップ関係にある同一システム内の別のノードを早急に修理してサービスを再開する必要がある。地

震等の大規模災害の影響を定量化することは難しいが、バックアップ中のノードが罹災する頻度を１０００年に１回以下、大規模災害の発生頻度を５０年に１回程度とすると、ノード間バックアップ率としては１／２０以下が必要となる。

また、これらのシステムに適用され、地理的に離れた地点で相互バックアップするような比較的規模の大きい装置の運用実績、開発状況から、平均故障間隔λ^{-1}はおおよそ１０００時間から５０００時間程度と想定し、ここでは、議論の煩雑化を避けるため、平均的な値としてλ^{-1}は２０００時間とした。

次に、装置の修理については、高度ＩＮやＯＬＴＰに適用される装置の修理状況等を考慮し、平均修理時間μ^{-1}は１時間とした。また、駆付け保守については、短時間で対応する緊急駆付けと長時間待たせてもよい非緊急駆付けの二種類を想定した。以下、非緊急駆付け時間をT_1、緊急駆付け時間をT_2とする。ここで、二重化された単一ノードの場合は、経験的に、駆付けの緊急度はシステムダウンしているか否か等により判断される場合が多い。しかし、二重化された２つのノードが更に相互にバックアップされる場合には、駆付け要求が発生する状態が多く、どのようなときに緊急駆付けとするかの判断が重要となる。

非緊急駆付けの時間T_1としては、①１～２日以内、②１週間程度、③１カ月以上、等が考えられる。保守チームの配備に着目すると、①の場合は、夜間は除くとしても、平日の昼間はもとより、土曜や日曜にも対応できる保守チームを配備する必要がある。②は平日勤務で、夜間や土、日曜は休みの保守チームで対応できる。また、①に比べて、駆付けに際し、時間的な余裕があるため、幾つかの障害を待たせる等のスケジュール化が可能であり、少ない保守チームで対応できる。③まで駆付け時間が延びた場合は、障害の発生を契機としてノードに駆付けるのではなく定期保守等で各ノードを巡回したときに、障害となっているモジュールを修理することで対応可能となる。少ない保守チームで、複数の障害に対応可能とするためには、１～２日以内では短すぎるため、本章では非緊急駆付け時間が１００時間以上許容されることを条件とする。

7.5　駆付けの緊急度と保守モデル

モジュールの障害が発生し駆付け要求が出される状態$S_{ab,cd}$（i=a、j=b、k=c、l=d で、ab,cd が 00,10／10,11／10,21／00,20／10,10／11,20／20,21／10.20／20,20 の場合）は9通りあり、どの程度緊急に駆付ける必要があるか否かの度合いに応じて、この9通りの状態をレベル分けすることを提案し、どのレベルまで緊急駆付けを行うかに基づいて保守モデルを設定する。本章では不稼働率の時間的な変化、すなわち$P_{SDN}(t)$に着目し、以下に示す方法により緊急度合いのレベル分けを行った。

駆付け要求が出される状態$S_{ab,cd}$の時刻 t ＝ 0 における確率$P_{ab,cd}(0)$を 1 とし、経過時間 t の関数として不稼働率$P_{SDN}(t)$を次式で算出する。

$$P_{SDN}(t) = P_{20,20}(t) + P_{20,21}(t) + P_{21,21}(t) \tag{7.23}$$

但し、$\omega_{ij,kl} = 0$ (7.24)

$$P_{ab,cd}(0) = 1 \tag{7.25}$$

$$P_{ij,kl}(0) = 0 \quad (i \neq a, j \neq b, k \neq c, l \neq d) \tag{7.26}$$

駆付け要求が出される9通りの状態について、 t と$P_{SDN}(t)$の関係を図7.4に示す。$P_{SDN}(t)$は式（7.1）～（7.16）を式（7.24）～（7.26）の条件のもとで数値計算した（付録7.1参照）。ここで、$\omega_{ij,kl}$を 0 としたのは、すべての駆付けが行われないとしたためである。図7.4で黒ぬりのマークを付した線はシステム内の2ノード共に保守チームがいない状態、白抜きのマークを付した線は片方のノードに既に保守チームが駆付けている状態を表している。 t が数時間以内では障害モジュール数（＝ a ＋ c ）が多いほど$P_{SDN}(t)$は高い。しかし、既に一方のノードに保守チームが駆付けている場合は、短時間のうちに当該ノードで修理が行われ t の増加と共に$P_{SDN}(t)$は小さくなる。 t が１０時間以上では、そのノードに障害モジュールが無かった場合と同様の傾向を示す。たとえば、$S_{20,21}$は4台のモジュールが障害となりシステムダウンしているが、既に一方のノードに保守チームが駆付けているため、短時間のうちに当該ノードで2台のモジュールの修理が行われ、 t の増加と共に$P_{SDN}(t)$は$S_{00,20}$と同様の値となる。図

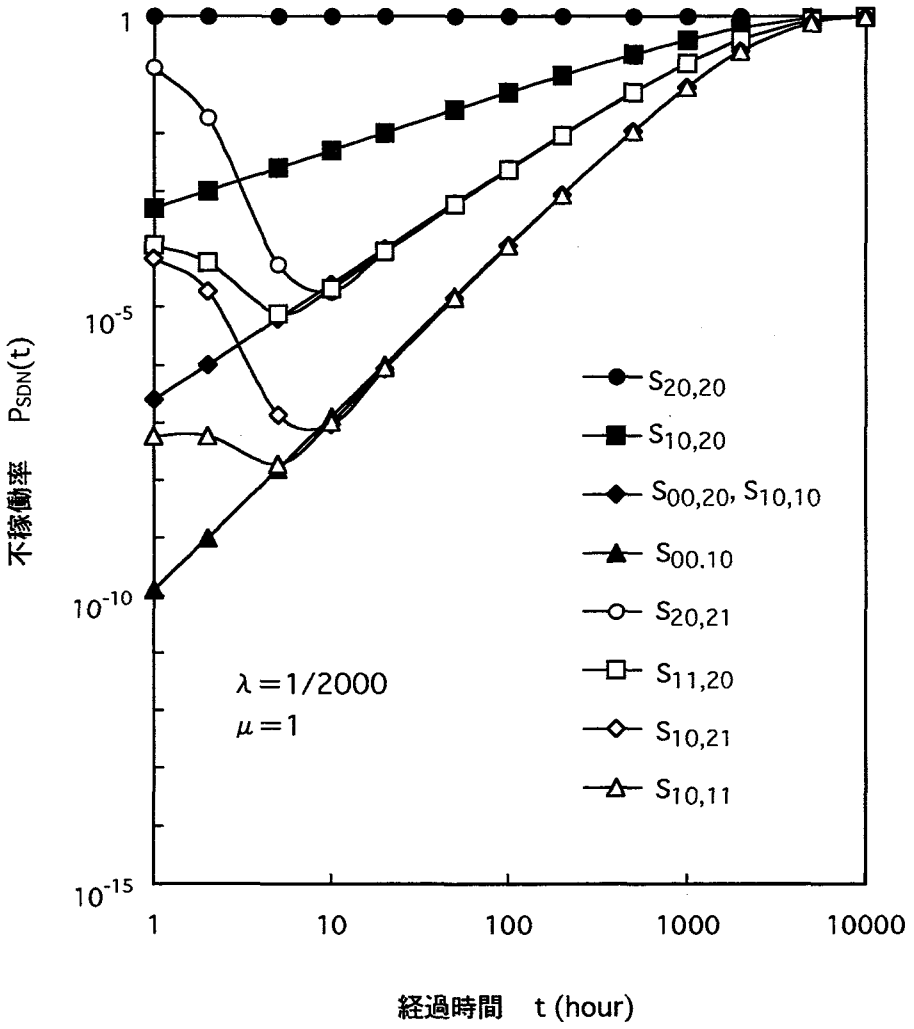


図７．４　経過時間と不稼働率

7.4から判るようにtが10時間以上では大きく4つにグループ化できる。同一の経過時間tで$P_{SDN}(t)$を比較すると、この大きさは、この状態で駆付け時間を長時間化していったときのリスクの大きさを表しており、駆付けの緊急度はこのリスクに対応すると考えられる。この尺度を緊急レベルHとし、図7.4に基づいて4段階に分けると、状態$S_{ab,cd}$の緊急レベルは次式で表される。

$$H=(a+c)-(a\cdot b+c\cdot d) \tag{7.27}$$

式（7.27）の第1項はシステム内の障害モジュール数、第2項は保守チームが到着しているノードでの障害モジュール数である（b、dが1の場合は保守チームが到着していることを示す）。すなわち、Hはシステム内で障害となっているが修理が始まっていないモジュール数となっている。

この緊急レベルHにより駆付け要求が出される9通りの状態を分類して図7.5に示す。緊急駆付けをどの状態で行うか否かについて、Hを基に3通りの保守モデル（以下、簡単にモデルと記す）を設定する。モデル0はHが4以上で緊急駆付けを行うモデルである。モデル1、モデル2はそれぞれHが3以上、Hが2以上で緊急駆付けを行うモデルである。図7.5にモデル毎にどの状態で緊急駆付けを行うかを併せて示す。また、駆付け要求が出される9通りの状態の駆付け時間$\omega_{ab,cd}^{-1}$として非緊急駆付け時間T_1と緊急駆付け時間T_2のどちらを設定するかを整理して表7.1に示す。

7.6　信頼性と保守方法の関係

7.6.1　システム信頼性の評価

要求されるシステム信頼性を満足した上で、保守方法に影響する緊急、非緊急駆付け時間として、どのような時間設定が可能であるか評価する。なお、システム信頼性は、はじめに表7.1に示した駆付け時間を式（7.1）～（7.16）に代入して、定常状

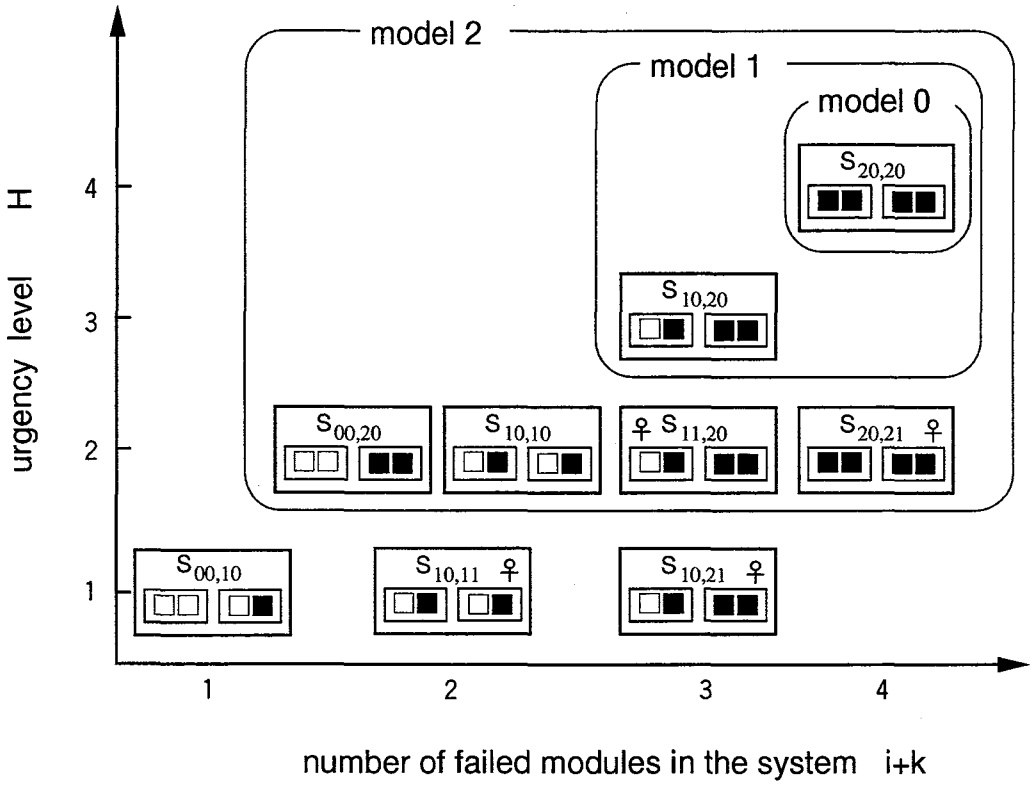


図7.5　緊急レベルと駆付けモデル

表7.1　駆付けモデルと設定する駆付け時間

緊急レベル	状態	駆付け時間	設定する駆付け時間		
			モデル0	モデル1	モデル2
4	$S_{20,20}$	$\omega_{20,20}^{-1}=$	T_2	T_2	T_2
3	$S_{10,20}$	$\omega_{10,20}^{-1}=$	T_1	T_2	T_2
2	$S_{20,21}$	$\omega_{20,21}^{-1}=$	T_1	T_1	T_2
	$S_{11,20}$	$\omega_{11,20}^{-1}=$	T_1	T_1	T_2
	$S_{10,10}$	$\omega_{10,10}^{-1}=$	T_1	T_1	T_2
	$S_{00,20}$	$\omega_{00,20}^{-1}=$	T_1	T_1	T_2
1	$S_{10,21}$	$\omega_{10,21}^{-1}=$	T_1	T_1	T_1
	$S_{10,11}$	$\omega_{10,11}^{-1}=$	T_1	T_1	T_1
	$S_{00,10}$	$\omega_{00,10}^{-1}=$	T_1	T_1	T_1

T_1：非緊急駆付け時間　　T_2：緊急駆付け時間

態における各状態の発生確率を数値計算し、次にそれらを式（7.17）～（7.22）に代入して算出した。

非緊急駆付け時間T_1と平均システムダウン時間T_{SDN}の関係を図７．６、緊急駆付け時間T_2と平均システムダウン時間の関係を図７．７に示す。平均システムダウン時間はT_1に係わらずほぼ一定となるが、T_2に対してはモデルに関係なく約T_2／２となる。平均システムダウン時間として１～３時間を条件としたため、T_2は２～６時間以下に抑える必要がある。以下、T_2は２～６時間として、T_1と不稼働率、デグレード率、平均デグレード時間、ノード間バックアップ率の関係について評価し、T_1として保守のスケジュール化が可能な１００時間以上許容されるか否かについて考察する。

T_1と不稼働率P_{SDN}の関係を図７．８に示す。Ｇｒａｙの信頼度分類によるクラス７、すなわち、不稼働率が10^{-7}以下を条件としたため、モデル０はT_1が４０～５０時間程度となり若干不足する。モデル１は１００～４００時間程度許容される。モデル２は非緊急駆付けを全く行わなくとも良く、過剰な保守方式となる。

T_1とデグレード率P_{DEG}の関係を図７．９に示す。デグレード率として不稼働率より３桁程度大きい10^{-4}以下を条件としたため、モデル０ではT_1は５０時間程度、モデル１では１００～２００時間程度許容される。モデル２では制約が無くなる。T_1と平均デグレード時間T_{DEG}の関係を図７．１０に示す。平均デグレード時間は、モデル１、２の場合は、T_1に依存せずほぼ一定となる。しかし、モデル０の場合は、T_1にほぼ比例して平均デグレード時間が長くなる。平均デグレード時間として１０時間程度を条件としたため、モデル０ではT_1に許容される時間は２０時間程度となり、モデル１、２では制約が無くなる。

T_1とノード間バックアップ率P_{BUP}の関係を図７．１１に示す。ノード間バックアップ率を１／２０以下とするためには、モデル０、１の場合はT_1を２００時間以下に抑える必要があるが、モデル２ではT_1への制約はなくなる。

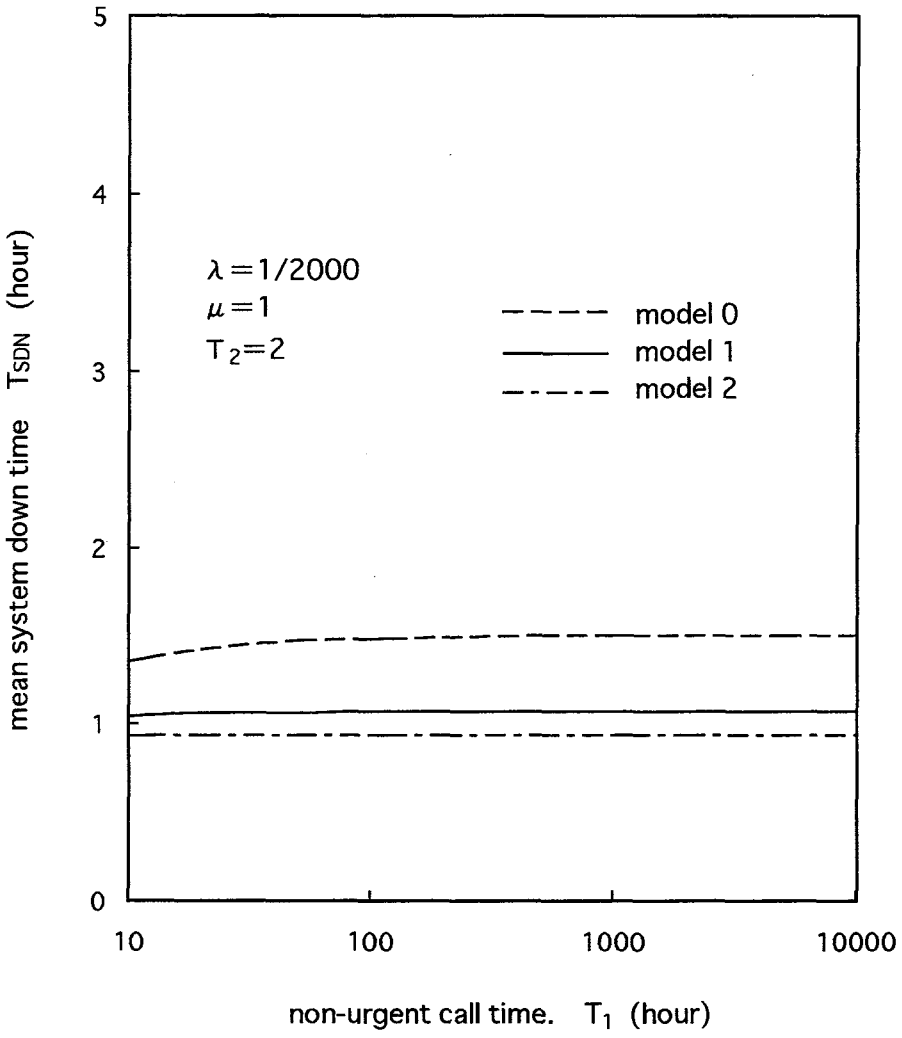


図７.６　非緊急駆付け時間と平均システムダウン時間

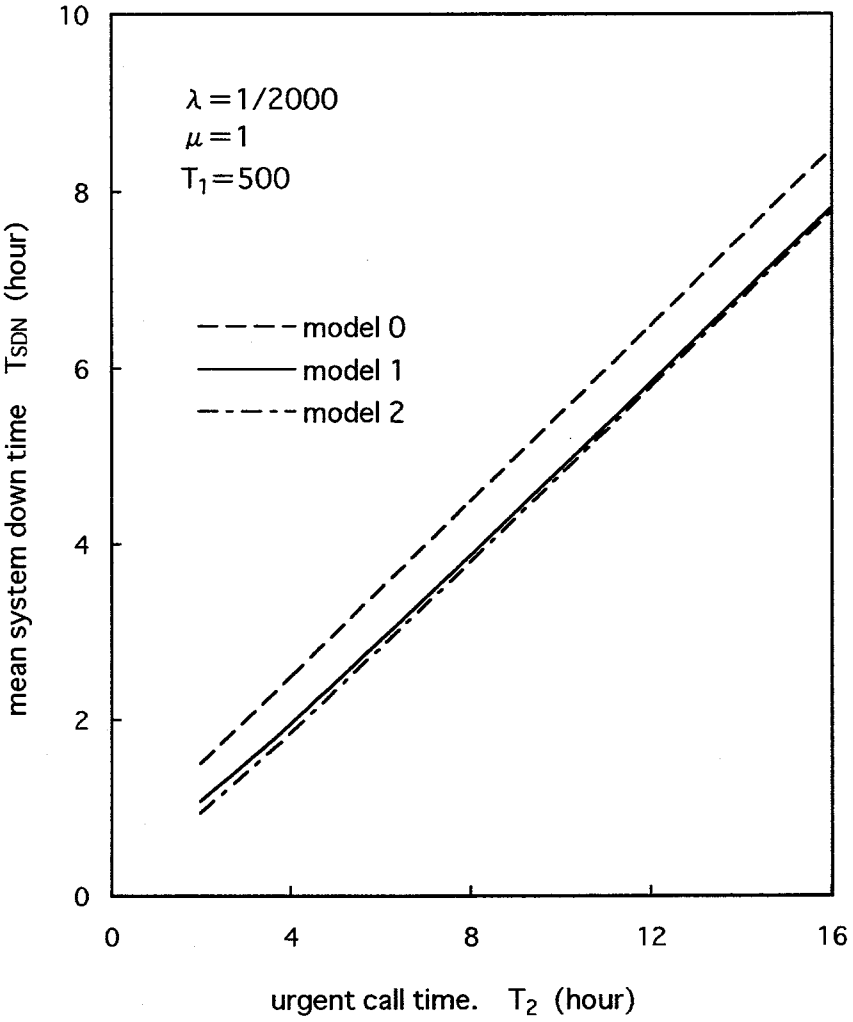


図7.7　緊急駆付け時間と平均システムダウン時間

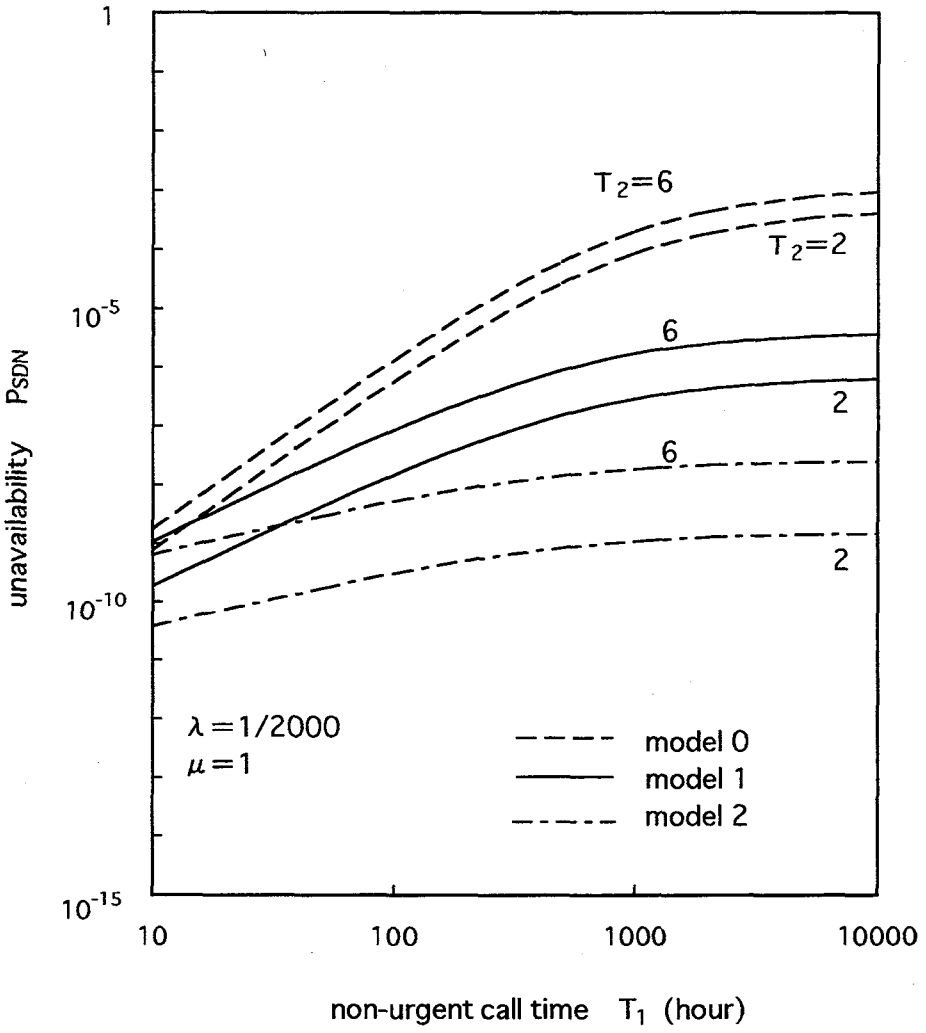


図７．８　非緊急駆付け時間と不稼働率

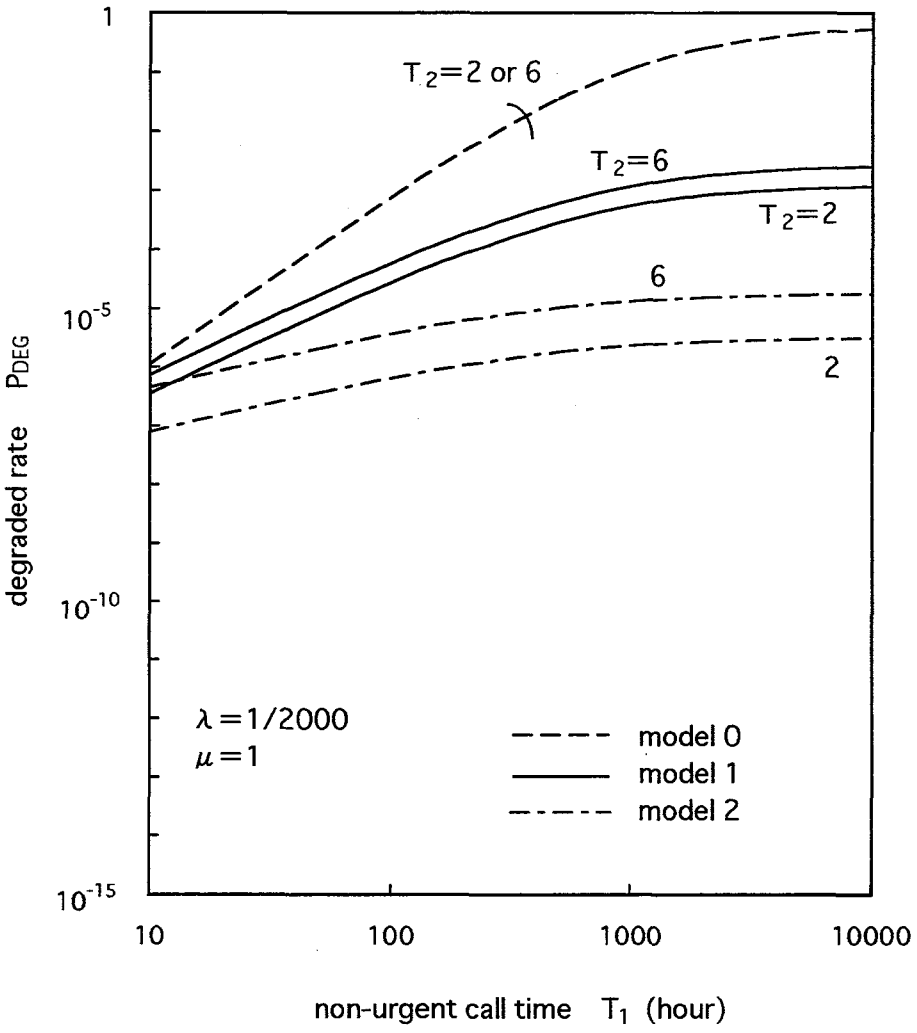


図７．９　非緊急駆付け時間とデグレード率

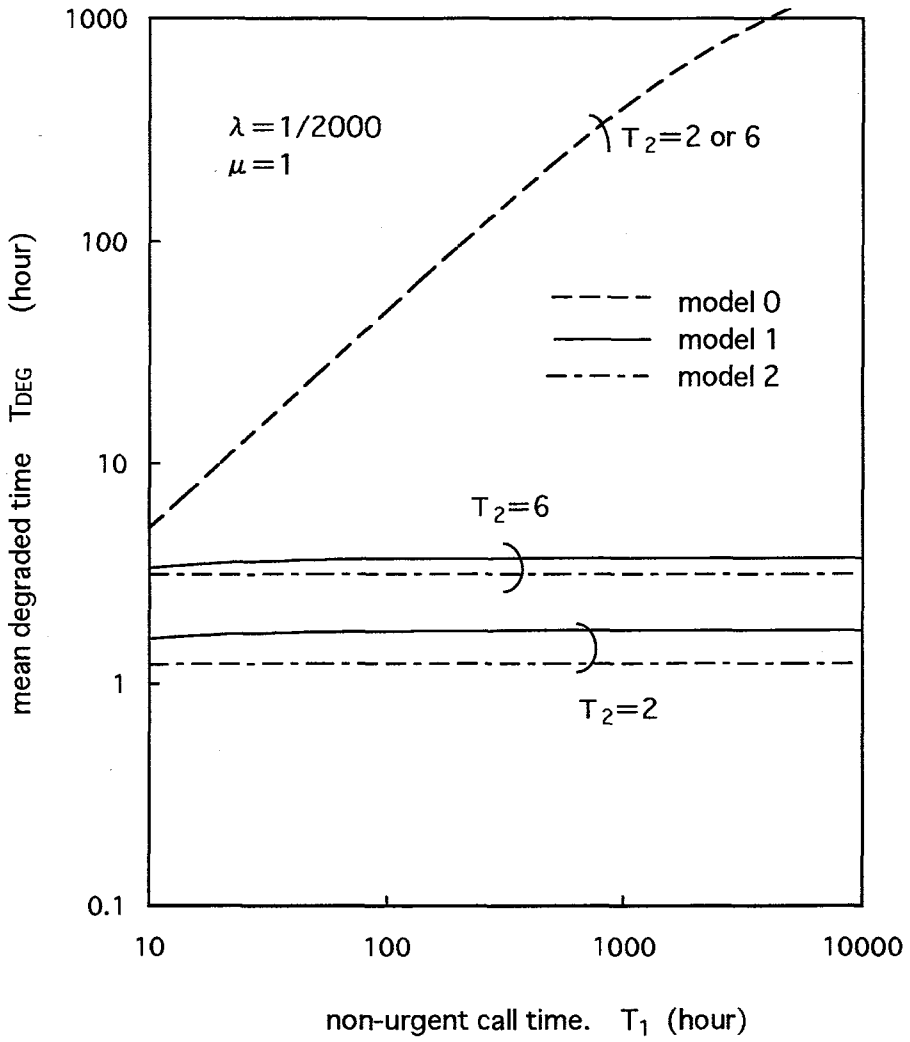


図７．１０　非緊急駆付け時間と平均デグレード時間

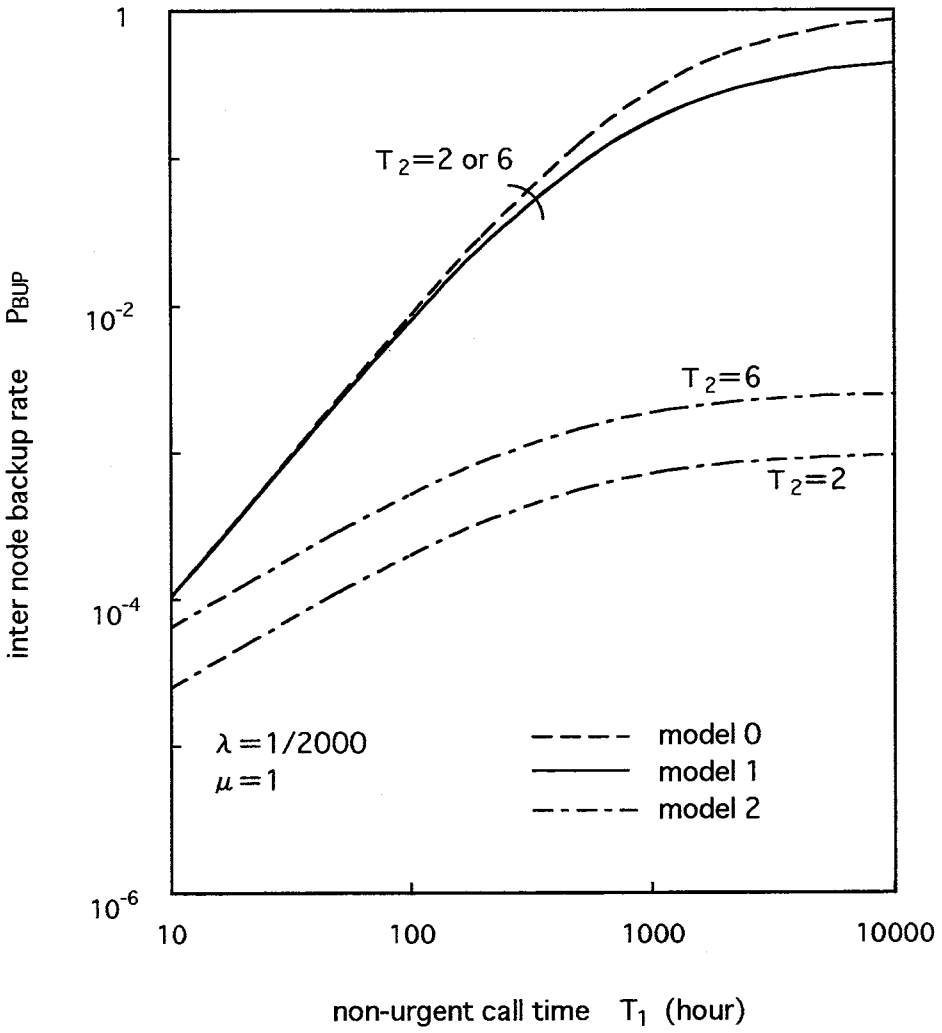


図7.11　非緊急駆付け時間とノード間バックアップ率

7.6.2 駆付け回数の評価

本節では、駆付け回数の面から各モデルの評価を行う。緊急駆付け要求が出される状態の確率の総和をP_{ECL}、非緊急駆付け要求が出される状態の確率の総和をP_{OCL}とすると、1年間で1つのシステムに対して発出される緊急駆付け回数N_{ECL}および非緊急駆付け回数N_{OCL}は、1年間の延べ時間が8760時間であることから、次式で表される。

$$N_{ECL} = 8760 \cdot P_{ECL} / T_2 \tag{7.28}$$

$$N_{OCL} = 8760 \cdot P_{OCL} / T_1 \tag{7.29}$$

ここで、総駆付け回数をN_{TCL}（$= N_{ECL} + N_{OCL}$）として、T_1とN_{TCL}の関係を図7.12に示す。図7.12の下部には、不稼働率を10^{-7}以下とした場合の各モデルのT_1の許容範囲についても併せて示している。保守のスケジュール化が可能と考えるT_1が100時間程度で、総駆付け回数を比較するとモデル間での差はほとんど無い。なお、総駆付け回数はモデルに関係なくT_1を長くすることにより減少する。これはT_1の増加につれて必ずしも1台のモジュールが障害となる毎に駆付けて修理するのではなく、2台のモジュールが障害となった時点で駆付ける場合が発生しているためである。

T_1と緊急、非緊急駆付け回数の関係を図7.13に示す。保守の効率化の点では緊急駆付け回数が少ない方が有利である。T_1が100時間で比べると、モデル2はシステムあたりの緊急駆付け回数が年に2回程度となり、モデル1とモデル0では0.1回程度となる。図7.12、図7.13では、T_2は2時間として評価したが、T_2を6時間としても緊急、非緊急駆付け回数は変化なかった。

7.6.3 保守形態の及ぼす影響

保守を効率化するポイントは緊急駆付け回数が少ないこと、非緊急駆付けについては保守のスケジュール化が可能と考えるT_1が100時間程度許容されることである。T_1として許容される時間や緊急駆付け回数について、前節までの評価結果をまとめ

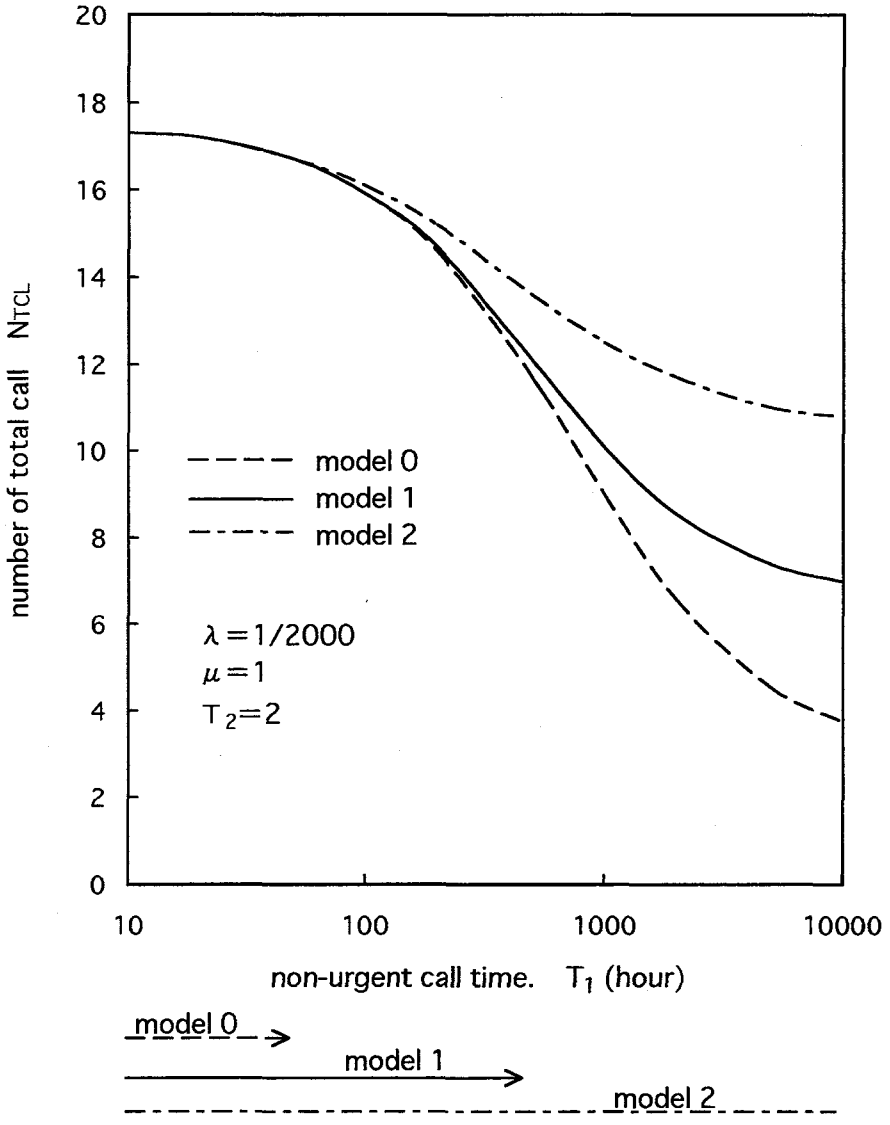


図７．１２　非緊急駆付け時間と総駆付け回数

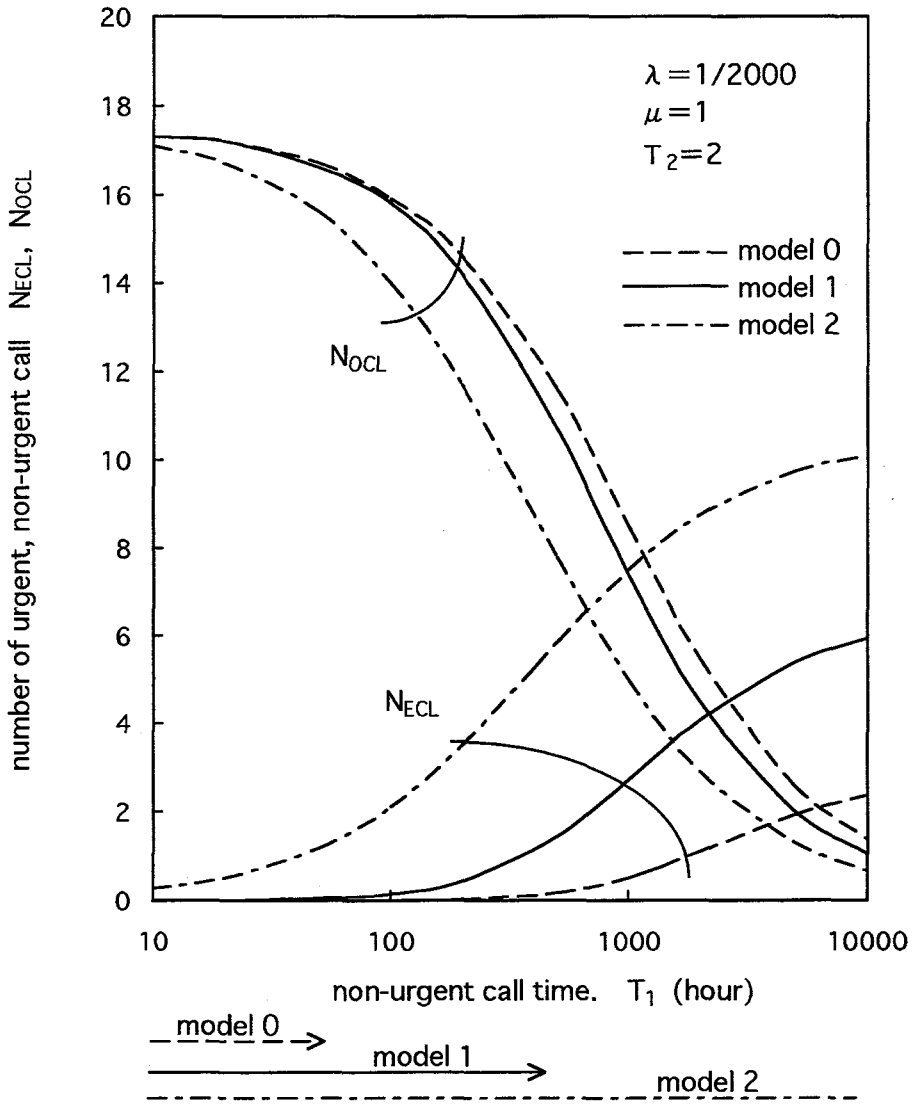


図７．１３　非緊急駆付け時間と緊急，非緊急駆付け回数

て表7.2に示す。表7.2より、モデル0は平均デグレード時間がT_1に比例して長くなるため、T_1を延ばすことは難しい。T_1として許容される時間は20時間程度と短く、保守のスケジュール化が十分行えない。モデル1ではT_1を100時間程度と大きく設定でき、保守のスケジュール化が十分可能である。モデル2は信頼性の面からは非緊急駆付け時間T_1への制約はないが、モデル1に比べて緊急駆付け回数が20倍程度多いという問題を持つ。

全体として、相互にバックアップした二重化ノードの保守としては、緊急レベル3以上で緊急駆付けを行うモデル1、すなわち、3～4台のモジュールが障害で、かつ両方のノードに保守チームがいない状態（$S_{10,20}$、$S_{20,20}$）のとき緊急駆付けを行うモデルが有用であると考える。モデル1の場合、すべてのモジュールが障害でシステムダウンしているが、既に一方のノードに保守チームが駆付けている状態（$S_{20,21}$）では緊急駆付けを行わない。この状態に対して、緊急駆付けを行うとしてもシステム信頼度の向上効果はほとんどなく、7.5節で提案した緊急レベルに基づくレベル分けが有効と言える。

7.6.4　システム、保守センタの配置方法

これまでの検討結果を踏まえ、システムや保守センタの配置方法について簡単に考察する。保守センタからノードまでは数時間以内の緊急駆付けを可能とする必要がある（以下、保守センタから緊急駆付け可能なエリアをセンタエリアと記す）。このためには、たとえば、車による駆付けを行うとした場合、センタエリアは保守センタからおおよそ100Km半径のエリア内となる。また、広域的な被害を被る地震、水害等でノードのペアが同時に被害を被らないためには、ノード間を数100Km以上離す必要がある。加えて、保守センタ数を極力少なくして、保守チームの集約化を図ることも必要である。これらの点を考慮したシステムおよび保守センタの配置の一例を図7.14に示す。図において、保守センタは2カ所に配置し、保守センタ相互間は数100Km以上離す。すべてのノードはいずれかのセンタエリア内に配置され、同

表７.２　駆付けモデルの評価結果

	設定条件	モデル０	モデル１	モデル２
非緊急駆付け時間(T_1)として許容される時間（時間）	平均システムダウン時間（1～3時間）	制限なし	制限なし	制限なし
	不稼働率（10^{-7}以下）	２０～４０	２００～４００	制限なし
	デグレード率（10^{-4}以下）	５０	１００～２００	制限なし
	デグレード時間（10時間）	２０	制限なし	制限なし
	ノード間バックアップ率（1/20以下）	２００	２００	制限なし
緊急駆付け回数（回/システム/年）		～０	０.１	２.０

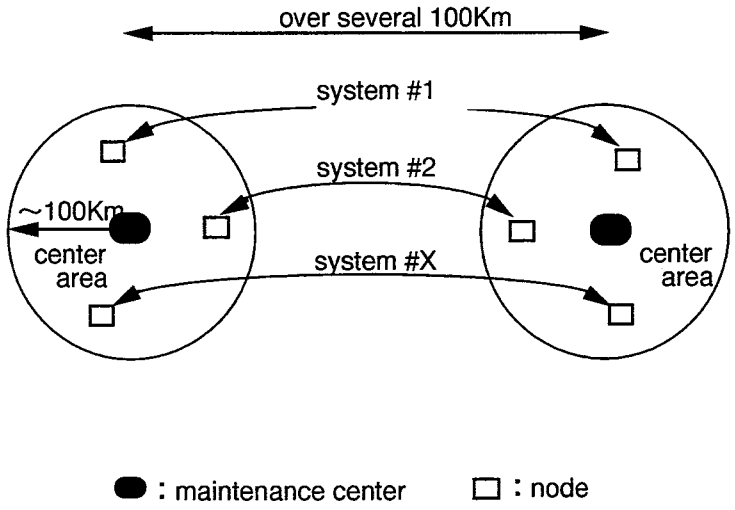


図７．１４　システム，保守センタの配置方法

—システム内のノードのペアはそれぞれ別のセンタエリア内に配置する。

7.7 結言

データベースを持ち、ノード内は二重化され、さらに2つのノード間で相互にバックアップされたシステム群を対象として、駆付け保守とシステム信頼性の関係について明らかにした。保守の方法は、障害が発生した時点で、システムの障害状態に応じて駆付ける方式とし、緊急、非緊急の二種類とした。このとき、障害発生時のシステム信頼度の時間的な変化に注目して、駆付けの緊急度合いをシステム内の障害装置数や保守チームが駆付けているか否かの状態に基づいてレベル分けすることを提案し、どのレベルまで緊急駆付けを行うかに応じて保守モデルを設定して信頼性や駆付け回数の評価を行った。これらの結果に基づいて、システムに要求される信頼度条件を満足し、非緊急駆付けに関しては保守のスケジュール化が可能な１００時間程度許容され、緊急駆付けに関しては駆付け回数の少ない効率的な保守方法を明らかにした。

高い信頼性が要求される高度ＩＮシステムや、ＯＬＴＰシステムでは地震や火災等に際してもサービスを継続するため、離れた地点での相互バックアップを行う方向にある。ここで示した保守の評価方法は、このバックアップを単に冗長度の増加としてとらえるのではなく、保守の効率化に活用するという面で、これら多くの高信頼化システムの保守方法を決定する上で有用である。

第8章　結論

本論文では、高い拡張性を持ち、高性能で高信頼なサービス制御ノードを実現するためのハードウェア構成技術や、全国的な高度ＩＮサービスを提供するためネットワーク内に多数設置されたサービス制御ノードの効率的な運転・保守技術について研究した。具体的には、需要の変動や機能要求へ柔軟に対応する分散処理化、サービス毎カスタマ毎のデータを効率的に管理するデータベース化、運転保守の効率化とのバランスを考慮した経済的な高信頼化の各面からアプローチし、（１）多数のモジュールによるデータベースの分散構成技術、（２）モジュール間の負荷の偏りの評価技術、（３）モジュール間の効率的な結合技術、（４）メモリデータベースの効率的なリカバリ技術、（５）多数のサービス制御ノードの経済的な運転保守技術、を明らかにした。また、試作を通して、提案した技術の有効性を確認した。本研究で得られた主な成果は以下の通りである。

（１）データベースの分散構成技術

データベースのアクセス特性、モジュールの負荷やその偏りなどを総合的に考慮して、データベースの分散構成方式の評価方法を明らかにした。これに基づき、サービス制御ノードとして効率的な分散型メモリデータベース構成法を提案した。

具体的には、呼処理に必要なサービスデータや、複数のＳＣＰにＳＬＰを分散配備したときの配備先を特定するＳＬＰディレクトリ等、サービス制御ノードとして必要なデータベースの容量やアクセス頻度から、データベースの格納媒体としてＯＬＴＰシステムで一般に用いられている磁気ディスク装置は適用できず、主メモリや半導体ディスク装置に格納する必要があることを示した。また、データベースの格納媒体や、複数のモジュールからのデータベースの共用の可否に基づいてデータベースの分散構成の代替案を設定し、モジュール間の負荷の偏り、データベース処理でのダイナミックステップ数、データベースの格納コストなどを総合的に考慮した評価方法を明ら

かにした。この結果に基づき、更新系データベースであるサービスデータについては、データベースを各モジュールに分散して配備し、主メモリに格納する分割方式を提案し、参照系データベースであるＳＬＰディレクトリについては全てのデータベースを各モジュールに重複して配備し、主メモリに格納する多重方式を提案した。データの高信頼化に関しては、ハードウェア障害やソフトウェアのバグによりデータが失われた場合のサービスへの影響度合いに基づく、データの復元条件を明らかにした。また、アベイラビリティの向上に関しては、モジュール間バックアップとノード間バックアップを組み合わせた効率的な冗長構成法を提案した。

分散型メモリデータベース構成と、モジュール間、ノード間での相互バックアップを組み合わせた冗長構成により、高性能で高信頼なサービス制御ノードを経済的に実現することができる。

（２）モジュール間の負荷の偏りの評価技術

カスタマ毎のサービスデータに対するトラヒックが大きくばらつく場合について、トラヒック量の近似方法を提案した。また、これを用いて、複数のモジュールにサービスデータを分散配備した場合のモジュール間の負荷の偏りの評価方法を明らかにした。

具体的には、最もトラヒックの高いカスタマと最もトラヒックの低いカスタマのトラヒック比が数千倍から数万倍と大きくばらつく場合について、各カスタマのトラヒックの高いものからの順位と、トラヒックのばらつきの度合いを表す偏り係数を用いて近似する方法を提案した。この近似方法は、高度ＩＮサービスのようにトラヒックの高い少数のカスタマが全体のトラヒックの大半を占めるような場合に良く近似できることを示した。これを用い、カスタマ毎のサービスデータを複数のモジュールに分散配備した場合のモジュール間の負荷の偏りの評価方法を明らかにした。この負荷の偏りは、モジュールに収容するサービスデータ数の平方根に反比例すること、偏り係数が１のとき極大値を取ることを示した。

また、高度ＩＮサービスのトランザクションの処理量や、現状のプロセッサ性能か

ら、モジュールに収容できるサービスデータ数は数万から１０万程度であり、このとき、モジュール間の負荷の偏りは１～２割程度に収まることを明らかにした。さらに、このモジュール間の負荷の偏りについては、ＳＬＰディレクトリ検索処理とサービスデータに基づくデータベース処理を組み合わせて、モジュール間の負荷を動的に平準化する方法を提案した。

負荷の偏りや平準化技術は分散処理によるサービス制御ノードの方式設計、設備設計に有効である。

（３）モジュール間の結合技術

大規模分散処理システムの結合機構の高信頼化構成法、通信処理能力を向上させる通信制御チャネルの制御方式およびモジュール間の通信時間の測定方法について提案した。

モジュール間の接続にはＡＴＭを適用し、結合機構を各モジュールと接続する機能、信号を多重化／分離化する機能、スイッチングを行う機能により実現し、機能単位に二重化し、接続するモジュール台数が５０モジュール程度とした大規模分散処理システムにおいても、システムの不稼働率を10^{-7}～10^{-6}以下にすることができた。各モジュールの通信制御については、ある周期毎にまとめて複数のメッセージを処理する方式を取り上げ、周期時間と通信時間の関係を明らかにし、高度ＩＮのＳＣＰでは数１０ミリ秒の周期で処理することが有効であることを示した。

ここで提案したモジュール間の結合技術は、高度ＩＮのサービス制御ノードに限らず、適用業務の拡大から多数のモジュールによる分散処理化が進んでいるＯＬＴＰシステムにおいても、高信頼で経済的な分散処理システムを構築する上で有用である。

（４）メモリデータベースのリカバリ技術

障害発生からデータベースが回復するまでの時間と、データベースの単位容量当たりのトラヒック量に着目し、メモリデータベースのリカバリ方式の評価方法を明らかにした。これに基づき、サービス制御ノードのリカバリ方式として、半導体ディスク

装置にメモリデータベースのログとチェックポイント時点でのデータベースを取得する効率的なリカバリ方式を提案した。

主メモリ上のデータベースが破壊された場合、データベースを復元するためには、あらかじめ復元に必要なログを外部の記憶装置に取得しておく必要がある。外部の記憶装置の形態としては、入出力による方法と通信による方法に大きく２大別され、それぞれを代表して、半導体ディスク装置を適用する方式と、別のモジュールの主メモリを適用する方式を代替案として設定した。この２つの代替案について、スループット、リカバリ時間に対する評価を行い、モジュールに収容可能なデータベース容量に着目し、両方式の適用領域を明らかにした。これに基づき、サービス制御ノードのメモリデータベースのリカバリ方式として、半導体ディスク装置にログとチェックポイント時点でのデータベースを取得する、効率的なリカバリ方式を明らかにした。また、半導体ディスク装置に取得したログ等の消失を防止するための不揮発化に関しては、直接バッテリでバックアップする方法と、データを磁気ディスク装置に移す方法を比較評価し、サービス制御ノードとして要求される２０時間程度のバックアップでは直接バックアップする方法がハードウェア量や、制御の簡単化の点で有利であることを示した。また、半導体ディスク装置をバックアップするバッテリについては、新たに劣化判定機構を開発し、保守の容易な方法を明らかにした。

メモリデータベースのリカバリ方式の評価方法は、サービス制御ノードに限らず、トランザクション処理の高性能化のためメモリデータベース化が進んでいる多くのＯＬＴＰシステムにおいても有効である。

（５）高信頼化運転保守技術

データベースを保有し、二重化された２つのノードを離れた地点に配置して相互にバックアップするシステムを対象とし、これらがネットワーク内に多数配置された場合について、駆付け保守の方法と信頼性の関係を明らかにし、高い信頼性と平均的に長い駆付け時間の双方を満足する効率的な運転保守方式を提案した。

保守チームは保守センタに集中配置され、システムの障害状態に応じて短時間で対

応する緊急駆付けと、長時間待たせてもよい非緊急駆付けの二種類の駆付けにより保守する方式を想定した。このとき、システムの信頼性の時間的な変化に注目し、駆付けの緊急度合いをシステム内の障害装置数や、保守チームが到着しているか否かの状態によりレベル分けし、このレベルにより駆付け保守の方法を設定することを提案した。これに基づいて、システムの信頼性や駆付け回数を総合的に評価し、システム内で３～４台のモジュールが障害で、かつ両方のノードに保守チームがいない状態のとき緊急駆付けを行い、それ以外の状態では複数の障害に対して保守のスケジュール化を行うなど非緊急に対応することが、保守の効率化の点で有効であることを明らかにした。さらに、この結果を踏まえ、サービス制御ノードや保守センタの配置方法を明らかにした。

高い信頼性が要求される高度ＩＮシステムや、ＯＬＴＰシステムでは地震や火災等に際してもサービスを継続するため、離れた地点での相互バックアップを行う方向にある。ここで示した保守の評価方法は、このバックアップを単に冗長度の増加としてとらえるのではなく、保守の効率化に活用するという面で、これら多くの高信頼化システムの保守方法を決定する上で有用である。

本研究により、拡張性のよい、高性能で、高信頼なサービス制御ノードを、経済的に構成することができる。また、多数のサービス制御ノードの効率的な運転保守が可能となる。本研究の成果を組み込んだ装置は、既に、１９９７年１１月からＰＨＳサービスやローミングサービスに使用されている[66]。引き続き、パーソナル通信サービス等、高度インテリジェントネットワークを用いた高機能な電話サービスに用いられる予定である。高度ＩＮサービスは、今後、多くの通信通信事業者により提供されることが予想されるが、本研究により提案した技術は通信事業者の区別なく適用できる。本研究が高度インテリジェントネットワークにおけるサービス制御ノードの構成方式や運転保守方式の進展の一助となれば幸いである。

参考文献

（1）S. Suzuki, “IN Rollout in Japan,” IEEE commun. mag., vol.31, pp.48-55, March 1993.

（2）E. Cancer, R. Mccann, and M. Aboudharam, “IN Rollout in Europe,” IEEE commun. mag., vol.31, pp.38-47, March 1993.

（3）P. A. Russo, K. Bechard, E. Brooks, R. L. Corn, R. Gove, W. L. Honig, and J. Young, “IN Rollout in the United States,” IEEE commun. mag., vol.31, pp.56-63, March 1993.

（4）鈴木滋彦, “高度インテリジェントネットワーク,” 電子情報通信学会誌, vol.77, no.4, pp.410-415, April 1994.

（5）平野正則, 鈴木孝至, 塩澤恒道, 芳西崇, 木ノ内康夫, “分散処理による高度ＩＮ用サービス制御ノードの構成,” 信学論(B-I), vol.J79-B-I, no.8, pp.539-550, Aug. 1996.

（6）石川弘, 石川秀樹, “ネットワークの高度化とサービスの展開,” ＮＴＴ技術ジャーナル, no.12, pp.4-8, Dec. 1989.

（7）関根俊彦, 佳山東一, 宮本雅昭, 渡部剛士, “ＮＳＰ／ＮＳＳＰのソフトウェア機能拡充,” ＮＴＴ技術ジャーナル, no.12, pp.70-72, Dec. 1994.

（8）鈴木滋彦, “新ノードシステムの開発,” ＮＴＴ Ｒ＆Ｄ, vol.45, no.6, pp.497-506, June 1996.

（9）上坂久一, 網谷駿介, “ＩＮサービスの先駆け　フリーダイヤルサービス,” ＮＴＴ技術ジャーナル, no.12, pp.21 -26, Dec. 1989.

（10）上坂久一, 吉江金三郎, 児玉博義, 吉見正信, “高度電話網サービス制御ソフトウェア構成法,” ＮＴＴ電気通信研究所研究実用化報告, vol.36, no.8, pp.1059-1064, Aug. 1987.

（11）村瀬節雄, 池尻稔, 武井伊佐夫, “サービスのカスタマ化とカスタマコントロール,” ＮＴＴ技術ジャーナル, no.12, pp.16-20, Dec. 1989.

（12）伊藤弘, “高度ＩＮのアーキテクチャ技術,” ＮＴＴ技術ジャーナル, no.6, pp.13

-17, June 1993.

(13) 鈴木滋彦, “インテリジェントネットワーク（ＩＮ）の高度化への展望,” ＮＴＴ技術ジャーナル, no.6, pp.8 -12, June 1993..

(14) 鈴木孝至, 上坂久一, 木ノ内康夫, “高度ＩＮのサービス制御技術,” ＮＴＴ技術ジャーナル, no.6, pp.18 -22, June 1993.

(15) 弓場英明, 佐藤清実, 今川仁, 鈴木孝至, 田中豪, “新ノード高度ＩＮ技術,” NTTR&D, vol.45, no.6, pp.559-568, June 1996.

(16) 佐藤清実, “高度ＩＮのサービス管理技術,” ＮＴＴ技術ジャーナル, no.6, pp.23 -37, June 1993.

(17) 花澤隆, “高度ＩＮのオペレーション技術,” ＮＴＴ技術ジャーナル, no.6, pp.28 -32, June 1993.

(18) M. T. Chao, and P. B. Passero, “Using General Purpose Computing Equipment as a Base for an Advanced Intelligent Network Systems Platform,” IEEE IN'95 Workshop, May 1995.

(19) M. Syrett, D. Skov, and A. Kristensen, “HP in IN,” IEEE IN'95 Workshop, May 1995.

(20) J. Gray, and A. Reuter, “TRANSACTION PROCESSING : CONCEPTS AND TECHNIQUES,” MORGAN KAUFMANN PUBLISHERS, CALIFORNIA, 1993.

(21) 渡辺栄一, Ｊ.グレイ, “オンライントランザクション処理：ＯＬＴＰシステム,” 近代科学社, 1995.

(22) 鶴保征城, 木ノ内康夫, 星子隆幸, 仲谷元, 宮川順治, “ループを用いた大規模分散処理システム,” 情報処理学会誌, vol.31, no.5, pp.686-697, May 1990.

(23) 竹内進, 宮崎達三, 貝沢哲男, 倉野明彦, “サービスを実現する高機能レイヤの信頼性を向上,” ＮＴＴ技術ジャーナル, no.9, pp.17-20, Sep. 1991.

(24) 浜田浩平, 黒住弘明, “リアルタイムネットワークシステム,” 信学誌, vol.73, no.11, pp.1161-1166, Nov. 1990.

(25) T. L. Casavant, and J. G. Kuhl, “A Taxonomy of Scheduling in General-Purpose Distributed Computing Systems,” IEEE Trans. on Software and Eng., vol.14, no.2,

pp.141-154, Feb. 1988.

(26) 渡辺尚, 太田剛, 水野忠則, 中西暉, "双方向ピギーバックに基づいた動的負荷分散方式,"信学論(D-I), vol.J78-D-I, no.3, pp.302-312, March 1995.

(27) 田中克之, "並列汎用機の全貌," 日系ＢＰ社, Sep. 1994.

(28) M.H. Eich, "Foreword Main Memory Databases : Current and Future Research Issues," IEEE Trans. on Knowl. and Data Eng., vol.4, no.6, pp.507-508, Dec. 1992.

(29) H. Garcia-Molina, and K. Salem, "Main Memory Database Systems : An Overview," IEEE Trans. on Knowl. and Data Eng., vol.4, no.6, pp.509-516, Dec. 1992.

(30) 泉谷建司, "ＥｔｈｅｒｎｅｔとＦＤＤＩ," pp.338-342, ソフト・リサーチ・センター, 1996.

(31) T. J. Lehman, and M. J. Carey, "A Recovery Algorithm for A High-Performance Memory-Resident Database System," Proc. ACM SIGMOD Int. Conf. on Management of Data, pp.104-117, May 1987.

(32) M.H. Eich, "Main Memory Database Recovery," ACM FJCC, pp.1226-1232, Nov. 1986.

(33) V. Kumar, and A. Burger, "Performance Measurement of Some Main Memory Database Recovery Algorithms," Proc. of 7th Int. Conf. on Data Engineering, pp.436-443, April 1991.

(34) 高倉弘喜, 上林彌彦, "フラッシュメモリバックアップ方式の設計と性能評価," 信学論(D-I), vol.J76-D-I, no.10, pp.514-521, Oct.1993.

(35) (財)金融情報システムセンター, "金融情報システム白書（平成９年版）," 財経詳報社, 1996.

(36) 渡辺均, 能條哲, "信頼性設計のためのシステム開発の動向," ＮＴＴ技術ジャーナル, no.3, pp.14 -18, March 1993.

(37) H. Watanabe, "A reliability design method for maintenance strategy of telecommunication network," Reliability and Maintainability Symposium, pp.476-483, Jan. 1993.

(38) M. Hirano, Y. Kinouchi, and T. Suzuki, "Distributed Control Node Architecture in the Advanced Intelligent Network," ISS '95, April 1995.

(39) 平野正則，塩澤恒道，木ノ内康夫，中村篤，井上潮，“分散データベースシステムのコストパフォーマンス評価,” 電子情報通信学会春季大会，D104，1992.

(40) 大光明直孝，林誠治，富上幸成，平野正則，“分散構成におけるデータベースのバックアップ方法,” 電子情報通信学会総合大会，D-6-22，1997.

(41) ITU-T 勧告，“Q.1200 シリーズの能力セット 1”，1993.

(42) 西原琢夫，富田清次，“蓄積情報を有するリアルタイムシステム構成法,”信学論(D-I)，vol.J78-D-I，no.8，pp.770-776，Aug. 1995.

(43) J. Gray, and D. P.Siewiorek，“High-Availability Computer Systems,” COMPUTER, pp.39-48, Sep. 1991.

(44) Technical Requirements TR100001，“Interface for Realtime Operating Systems,” Ver.1, NTT，1992.

(45) 大久保利一，香西省治，二神新，大南正人，“リアルタイム性を考慮した通信処理用ＯＳインタフェースの設計と性能評価,” 信学論(D-I)，Vol.J78-D-I，No.8，pp.687-698，Aug. 1995.

(46) 平野正則，塩澤恒道，木ノ内康夫，鈴木孝至，“高度ＩＮの分散処理におけるデータ配置による負荷の偏り,” 信学論(B-I)，vol.J-80-B-I，no.2，pp.87-97，Feb. 1997.

(47) M. Hirano, T. Shiozawa, Y. Kinouchi, and T. Suzuki, “Large-Scale Distributed Control Node in the Advanced Intelligent Network,” APCC '95，June 1995.

(48) 富上幸成，平野正則，大光明直孝，吉見正信，木ノ内康夫，“高度ＩＮにおける分散データベース処理の負荷平準化に関する一考察,” 電子情報通信学会通信ソサイエティ大会，B-623，1996.

(49) 水谷静夫，“朝倉日本語新講座 2 語彙,” pp.102-105，朝倉書店，1983.

(50) 小杉肇，“統計調査論,” pp.128-129，朝倉書店，1976.

(51) 平野正則，櫻井秀紀，今川仁，木ノ内康夫，“分散処理による高度ＩＮ用サービス制御ノードのモジュール間結合方式,” 信学論(B-I)，vol.J81-B-I，no.8，pp.519-530，Aug. 1998.

(52) H. Sakurai, M. Ito, M. Hirano, and H.Imagawa, "Inter-module Connections in Distributed-processing Service Control Points in the Advanced Intelligent Network and Testing of Associated Hardware," ICIN '96, Nov. 1996.

(53) 櫻井秀紀，塩澤恒道，平野正則，今川仁，"ＡＴＭ結合機構を用いた分散処理システムの試験方式," 電子情報通信学会通信ソサイエティ大会，B-406，1995.

(54) 斉藤秀一，大光明直孝，櫻井秀紀，伊藤守夫，平野正則，"分散処理による高度ＩＮ用サービス制御ノードのモジュール間結合方式に関する一考察," 電子情報通信学会総合大会，B-695，1996.

(55) 脇村慶明，"ディジタル交換システム," 信学誌，vol.73，no.11，pp.1167-1173，Nov. 1990.

(56) S. Heatley, and D. Stokesberry, "Analysis of Transport Measurements Over a Local Area Network," IEEE Commun. Mag., vol.27, pp.16-22, June 1989.

(57) S. Nojo, and H. Watanabe, "Incorporating reliability specifications in the design of telecommunication networks," IEEE Commun. Mag., vol.31, pp.40-43, June 1993.

(58) 平野正則，山根道広，山崎幹夫，木ノ内康夫，林　誠治，"高度ＩＮ用サービス制御ノードにおけるメモリデータベースのリカバリ方式," 信学論(B-I)，vol.J80-B-I，no.8，pp.596-608，Aug. 1997.

(59) 林誠治，平野正則，石橋宏純，小林伸幸，白石正裕，"高度ＩＮ用データベースのリカバリ方式に関する一考察," 電子情報通信学会総合大会，B-658，1996.

(60) 斉藤秀一，伊藤守夫，平野正則，櫻井秀紀，林誠治，"高度ＩＮ用半導体ファイル装置のバッテリバックアップ方法," 電子情報通信学会総合大会，B-6-118，1997.

(61) 川瀬克之，秋葉昭浩，大光明直孝，平野正則，"高度ＩＮ用大容量半導体ファイル装置のメモリ試験に関する一検討," 電子情報通信学会総合大会，B-6-4，1998.

(62) 山根道広，秋葉昭浩，平野正則，"半導体ファイル用バッテリバックアップ電源の高信頼化構成," 電子情報通信学会総合大会，B-6-66，1998.

(63) K. Takeno, M. Yamasaki, and S. Muroyama, "A New Backup Power Supply with a Battery Deterioration Test Circuit," INTELEC'95, pp.591-596, Oct. 1995.

(64) 平野正則，山根道広，小林正光，木ノ内康夫，"相互にバックアップされたデータベースを持つ二重化ノードにおける駆付け保守の評価," 信学論(B-I), vol.J81-B-I, no.10, pp.***-***, Oct. 1998.

(65) 山根道広，平野正則，木ノ内康夫，"センタ間バックアップを行う高信頼化システムの平均修理時間," 電子情報通信学会春季大会，A-389，1994.

(66) T. Hanazawa, "NTT's Intelligent Network Today and Tomorrow," ICIN '98, May 1998.

付録

第4章

付録4.1：トラヒックの期待値の算出

$$y_0 = \int_0^1 y dz = \int_0^1 c\left\{(g^{\frac{1}{\alpha}} - 1)z + 1\right\}^{-\alpha} dz \qquad (A\cdot4.1.1)$$

ここで、

$$z = \frac{x-1}{h-1} \qquad (A\cdot4.1.2)$$

$$h = g^{\frac{1}{\alpha}} \qquad (A\cdot4.1.3)$$

とおき、式（A·4.1.2）、（A·4.1.3）を式（A·4.1.1）に代入し次式を得る。

$$y_0 = \frac{c}{h-1}\int_1^h x^{-\alpha} dx \qquad (A\cdot4.1.4)$$

（1）$\alpha \neq 1$の場合

式（A·4.1.4）から次式を得る。

$$y_0 = \frac{c}{h-1}\left[\frac{x^{1-\alpha}}{1-\alpha}\right]_1^h = \frac{c(h^{1-\alpha} - 1)}{(1-\alpha)(h-1)} \qquad (A\cdot4.1.5)$$

式（A·4.1.5）に式（A·4.1.3）を代入し、次式を得る。

$$y_0 = \frac{c(g^{\frac{1}{\alpha}-1} - 1)}{(1-\alpha)(g^{\frac{1}{\alpha}} - 1)} \qquad (A\cdot4.1.6)$$

（2）$\alpha = 1$の場合

式（A·4.1.4）から次式を得る。

$$y_0 = \frac{c}{h-1}\left[L_{og}x\right]_1^h = \frac{cL_{og}h}{(h-1)} \qquad (A\cdot4.1.7)$$

式（A·4.1.7）に式（A·4.1.3）を代入し、次式を得る。

$$y_0 = \frac{cL_{og}g}{(g-1)} \qquad (A\cdot4.1.8)$$

付録４.２：トラヒックの累積値の期待値に対する割合の算出

$$\eta = \frac{\int_0^z y dz}{y_0} \qquad (A\cdot4.2.1)$$

ここで、

$$\int_0^z y dz = \int_0^z c\left\{(g^{\frac{1}{\alpha}}-1)z+1\right\}^{-\alpha} dz \qquad (A\cdot4.2.2)$$

に、式（A·4.1.2）、（A·4.1.3）を代入し、次式を得る。

$$\int_0^z y dz = \frac{1}{h-1}\int_1^{(h-1)z+1} cx^{-\alpha} dx$$

$$= \frac{c((hz-z+1)^{1-\alpha}-1)}{(1-\alpha)(h-1)} \qquad (A\cdot4.2.3)$$

式（A·4.2.3）に式（A·4.1.3）を代入し次式を得る。

$$\int_0^z y dz = \frac{c((g^{\frac{1}{\alpha}}z-z+1)^{1-\alpha}-1)}{(1-\alpha)(g^{\frac{1}{\alpha}}-1)} \qquad (A\cdot4.2.4)$$

式（A·4.2.4）、式（A·4.1.6）を式（A·4.2.1）に代入し、次式を得る。

$$\eta = \frac{(z-zg^{-\frac{1}{\alpha}}+g^{-\frac{1}{\alpha}})^{1-\alpha}-g^{1-\frac{1}{\alpha}}}{(1-g^{1-\frac{1}{\alpha}})} \qquad (A\cdot4.2.5)$$

付録４.３：αを無限大としたときのトラヒック量の算出

$$y = c\left\{(g^{\frac{1}{\alpha}} - 1)z + 1\right\}^{-\alpha} \tag{A·4.3.1}$$

ｙはｚの関数であり、ｚ＝０のまわりでテーラ展開すると、次式で表せる。

$$y(z) = y(0) + y'(0)\frac{z}{1!} + y''(0)\frac{z^2}{2!} + \cdots + y^{(n)}(0)\frac{z^n}{n!} + \cdots \tag{A·4.3.2}$$

また、式（A·4.3.1）より次式を得る。

$$y^{(n)}(z) = c(-1)^n\alpha(\alpha+1)\cdots(\alpha+n-1)(g^{\frac{1}{\alpha}} - 1)^n\left\{(g^{\frac{1}{\alpha}} - 1)z + 1\right\}^{-(\alpha+n)} \tag{A·4.3.3}$$

式（A·4.3.1）、式（A·4.3.3）でｚ＝０とおき、

$$y(0) = c \tag{A·4.3.4}$$

$$y^{(n)}(0) = c(-1)^n\alpha(\alpha+1)\cdots(\alpha+n-1)(g^{\frac{1}{\alpha}} - 1)^n \tag{A·4.3.5}$$

式（A·4.3.4）、式（A·4.3.5）を式（A·4.3.2）に代入して、$\alpha \to \infty$とおくと次式を得る。

$$\lim_{\alpha\to\infty} y(z) = c\left(1 + \sum_{n=0}^{\infty}\left\{\lim_{\alpha\to\infty}(-1)^n\alpha(\alpha+1)\cdots(\alpha+n-1)(g^{\frac{1}{\alpha}} - 1)^n\right\}\frac{z^n}{n!}\right) \tag{A·4.3.6}$$

ここで、ロピタルの規則を適用し、次式が成り立つ。

$$\lim_{\alpha\to\infty}\alpha^n(g^{\frac{1}{\alpha}} - 1)^n = \left\{\lim_{\alpha\to\infty}\alpha(g^{\frac{1}{\alpha}} - 1)\right\}^n = (L_{og}g)^n \tag{A·4.3.7}$$

また、ｉ＜ｎの場合、次式が成り立つ。

$$\lim_{\alpha\to\infty}\alpha^i(g^{\frac{1}{\alpha}} - 1)^n = \left\{\lim_{\alpha\to\infty}\alpha(g^{\frac{1}{\alpha}} - 1)\right\}^i \lim_{\alpha\to\infty}(g^{\frac{1}{\alpha}} - 1)^{n-i} = 0 \tag{A·4.3.8}$$

式（A·4.3.7）、式（A·4.3.8）を式（A·4.3.6）に代入し、次式を得る。

$$\lim_{\alpha\to\infty} y(z) = c\left(1 + \sum_{n=0}^{\infty}(-L_{og}g)^n\frac{z^n}{n!}\right) = ce^{-(L_{og}g)z} = cg^{-z} \tag{A·4.3.9}$$

付録４.４：負荷の偏りの標準偏差の算出

$$\sigma = \left\{ \frac{1}{(h-1)^R \lambda_0^2} \int_1^h \cdots \int_1^h (cx_1^{-\alpha} + \cdots + cx_R^{-\alpha} - \lambda_0)^2 dx_1 \cdots dx_R \right\}^{1/2} \qquad (A\cdot 4.4.1)$$

ここで、

$$\begin{aligned} H &= (cx_1^{-\alpha} + \cdots + cx_R^{-\alpha} - \lambda_0)^2 \\ &= c^2(x_1^{-2\alpha} + x_2^{-2\alpha} + \cdots + x_R^{-2\alpha}) + 2c^2(x_1^{-\alpha}x_2^{-\alpha} + \cdots + x_{R-1}^{-\alpha}x_R^{-\alpha}) \\ &\quad -2c\lambda_0(x_1^{-\alpha} + x_2^{-\alpha} + \cdots + x_R^{-\alpha}) + \lambda_0^2 \end{aligned} \qquad (A\cdot 4.4.2)$$

とおき、各項毎に積分を行う。

【$x_i^{-2\alpha}$の項の積分】

$$\int_1^h \cdots \int_1^h x_i^{-2\alpha} \cdot dx_R \cdots dx_R = \frac{(h^{1-2\alpha} - 1)(h-1)^{R-1}}{(1-2\alpha)} \qquad (A\cdot 4.4.3)$$

【$x_i^{-\alpha}x_j^{-\alpha}$の項の積分】

$$\int_1^h \cdots \int_1^h x_i^{-\alpha}x_j^{-\alpha} dx_1 \cdots dx_R = \frac{(h^{1-\alpha} - 1)^2(h-1)^{R-2}}{(1-\alpha)^2} \qquad (A\cdot 4.4.4)$$

【$x_i^{-\alpha}$の項の積分】

$$\int_1^h \cdots \int_1^h x_i^{-\alpha} dx_1 \cdots dx_R = \frac{(h^{1-\alpha} - 1)(h-1)^{R-1}}{(1-\alpha)} \qquad (A\cdot 4.4.5)$$

【定数項の積分】

$$\int_1^h \cdots \int_1^h dx_1 \cdots dx_R = (h-1)^R \qquad (A\cdot 4.4.6)$$

式（A·4.4.3）～式（A·4.4.6）式を式（A·4.4.1）に代入し、次式を得る。

$$\sigma = \frac{1}{(h-1)^{R/2}\lambda_0}\left\{\frac{c^2R(h^{1-2\alpha}-1)(h-1)^{R-1}}{(1-2\alpha)}\right.$$
$$+\frac{c^2R(R-1)(h^{1-\alpha}-1)^2(h-1)^{R-2}}{(1-\alpha)^2} - \frac{2cR\lambda_0(h^{1-\alpha}-1)(h-1)^{R-1}}{(1-\alpha)}$$
$$\left.+\lambda_0^2(h-1)^R\right\}^{1/2} \qquad (A\cdot4.4.7)$$

$\lambda_0 = y_0R$及び本文中の式（4.5）を代入し、次式を得る。

$$\sigma = \frac{1}{\sqrt{R}}\left\{\frac{(1-\alpha)^2(h-1)(h^{1-2\alpha}-1)}{(1-2\alpha)(h^{1-\alpha}-1)^2}-1\right\}^{1/2}$$
$$= \frac{1}{\sqrt{R}}\left\{\frac{(1-\alpha)^2(g^{\frac{1}{\alpha}}-1)(g^{\frac{1}{\alpha}-2}-1)}{(1-2\alpha)(g^{\frac{1}{\alpha}-1}-1)^2}-1\right\}^{1/2} \qquad (A\cdot4.4.8)$$

付録４.５：α＝１でＡが極大値を取ることの証明

$$A(\alpha,g) = \sqrt{\left\{\frac{(1-\alpha)^2(g^{\frac{1}{\alpha}}-1)(g^{\frac{1}{\alpha}-2}-1)}{(1-2\alpha)(g^{\frac{1}{\alpha}-1}-1)^2}-1\right\}} \qquad (A\cdot4.5.1)$$

$A(\alpha,g)$をαで微分すると次式を得る。

$$\frac{dA(\alpha,g)}{d\alpha} = \frac{1}{2A}\left\{\frac{(2\alpha^2(g^2-g^{\frac{1}{\alpha}})(1-g^{\frac{1}{\alpha}})+(1-2\alpha)(g^2+1-2g^{\frac{1}{\alpha}})g^{\frac{1}{\alpha}}\log g)(\alpha-1)^2}{\alpha^2(2\alpha-1)^2(g-g^{\frac{1}{\alpha}})^2}+\right.$$
$$\left.\frac{2(g^2-g^{\frac{1}{\alpha}})(g^{\frac{1}{\alpha}}-1)(\alpha-1)(g\alpha^2-g^{\frac{1}{\alpha}}\alpha^2+g^{\frac{1}{\alpha}}\log g-g^{\frac{1}{\alpha}}\alpha\log g)}{\alpha^2(2\alpha-1)(g-g^{\frac{1}{\alpha}})^3}\right\} \qquad (A\cdot4.5.2)$$

ここで、$\alpha \to 1$とすると次式を得る。

$$\lim_{\alpha\to 1}\frac{dA(\alpha,g)}{d\alpha}=\frac{1}{2A}\left\{-g(g-1)^2(2+\log g)\lim_{\alpha\to 1}\frac{(\alpha-1)^2}{(g-g^{\frac{1}{\alpha}})^2}+\right.$$

$$\left.2g(g-1)^2\lim_{\alpha\to 1}\frac{(\alpha-1)(g\alpha^2-g^{\frac{1}{\alpha}}\alpha^2+g^{\frac{1}{\alpha}}\log g-g^{\frac{1}{\alpha}}\alpha\log g)}{(g-g^{\frac{1}{\alpha}})^3}\right\} \qquad \text{(A・4.5.3)}$$

ここで、ロピタルの規則を適用し次式が成り立つ。

$$V=\lim_{\alpha\to 1}\frac{(\alpha-1)^2}{(g-g^{\frac{1}{\alpha}})^2}=\left\{\lim_{\alpha\to 1}\frac{\frac{d}{d\alpha}(\alpha-1)}{\frac{d}{d\alpha}(g-g^{\frac{1}{\alpha}})}\right\}^2=\left\{\lim_{\alpha\to 1}\frac{1}{\frac{gL_{og}g}{\alpha^2}}\right\}^2=\frac{1}{g^2(L_{og}g)^2} \qquad \text{(A・4.5.4)}$$

$$W=\lim_{\alpha\to 1}\frac{(\alpha-1)(g\alpha^2-g^{\frac{1}{\alpha}}\alpha^2+g^{\frac{1}{\alpha}}L_{og}g-g^{\frac{1}{\alpha}}\alpha L_{og}g)}{(g-g^{\frac{1}{\alpha}})^3}$$

$$=\frac{\lim_{\alpha\to 1}d^3(\alpha-1)(g\alpha^2-g^{\frac{1}{\alpha}}\alpha^2+g^{\frac{1}{\alpha}}L_{og}g-g^{\frac{1}{\alpha}}\alpha L_{og}g)/d\alpha^3}{\lim_{\alpha\to 1}d^3(g-g^{\frac{1}{\alpha}})^3/d\alpha^3}$$

$$=\frac{3gL_{og}g(2+L_{og}g)}{6g^3(L_{og}g)^2}=\frac{2+L_{og}g}{2g^2(L_{og}g)^2} \qquad \text{(A・4.5.5)}$$

なお、Wの分子、分母の３回微分の値は以下の通りである。

$$\text{Wの分母}=\lim_{\alpha\to 1}\left\{\frac{6g^{\frac{3}{\alpha}}(L_{og}g)^3}{\alpha^3}-\frac{18(g-g^{\frac{1}{\alpha}})g^{\frac{1}{\alpha}}(L_{og}g)^2(2\alpha+L_{og}g)}{\alpha^6}\right.$$

$$\left.+\frac{3(g-g^{\frac{1}{\alpha}})^2g^{\frac{1}{\alpha}}L_{og}g(6\alpha^2+6\alpha L_{og}g+(L_{og}g)^2}{\alpha^6}\right\}$$

$$=6g^3(L_{og}g)^3 \qquad \text{(A・4.5.6)}$$

$$\text{Wの分子} = \lim_{\alpha \to 1} \left\{ 6(g - g^{\frac{1}{\alpha}}) + \frac{3g^{\frac{1}{\alpha}} L_{0g} g}{\alpha^4} (2\alpha^3 + \alpha(2-\alpha) L_{og} g + (1-\alpha)(L_{og} g)^2) \right.$$

$$\left. + \frac{(\alpha - 1) g^{\frac{1}{\alpha}} (L_{og} g)^2}{\alpha^6} (4\alpha^2 L_{og} g - 6\alpha^2 - 6\alpha L_{og} g + (\alpha - 1)(L_{og} g)^2) \right\}$$

$$= 3 g L_{og} g (2 + L_{og} g) \qquad \text{(A·4.5.7)}$$

式（A·4.5.4）、式（A·4.5.5）を式（A·4.5.2）に代入し、次式を得る。

$$\lim_{\alpha \to 1} \frac{dA(\alpha, g)}{d\alpha} = 0 \qquad \text{(A·4.5.8)}$$

第５章

付録５.１：X（h）の算出

周期ｉで発生するメッセージ数をn^i、その発生確率を$P^i(n^i)$、周期ｉでの処理残りメッセージ数をh^i、その発生確率を$X^i(h^i)$、周期内で処理可能な最大メッセージ数をｂとおく。周期ｉ＋１での処理残りメッセージ数h^{i+1}は次式で表せる。

$$h^{i+1} = \begin{cases} n^i + h^i - b & (n^i + h^i \geq b) \\ 0 & (n^i + h^i < b) \end{cases} \qquad \text{(A·5.1.1)}$$

周期ｉでh^i個のメッセージが残る確率$X^i(h^i)$、とn^i個のメッセージが発生する確率$P^i(n^i)$は独立であり、周期ｉ＋１でh^{i+1}個のメッセージが残る確率$X^{i+1}(h^{i+1})$は、上記関係式が成立するh^iとn^iのすべての組合わせの確率$X^i(h^i) \cdot P^i(n^i)$の総和となり、次式で表せる。

$$X^{i+1}(1) = X^i(0)\cdot P(b+1) + X^i(1)\cdot P(b) + \cdots$$
$$\cdots + X^i(b)\cdot P(1) + X^i(b+1)\cdot P(0) \qquad (A\cdot 5.1.2)$$
$$X^{i+1}(2) = X^i(0)\cdot P(b+2) + X^i(1)\cdot P(b+1) + \cdots$$
$$\cdots + X^i(b+1)\cdot P(1) + X^i(b+2)\cdot P(0) \qquad (A\cdot 5.1.3)$$
$$\cdots\cdots\cdots$$

また、$X^{i+1}(h^{i+1})$の総和は１となることから、

$$X^{i+1}(0) = 1 - X^{i+1}(1) - X^{i+1}(2) - X^{i+1}(3) - \cdots \qquad (A\cdot 5.1.4)$$

となる。ここで初期値は、

$$X^0(0) = 1 \qquad (A\cdot 5.1.5)$$
$$X^0(1) + X^0(2) + X^0(3) + \cdots = 0 \qquad (A\cdot 5.1.6)$$

である。式（A･5.1.5）、（A･5.1.6）を式（A･5.1.2）～（A･5.1.4）の漸化式に代入し、ｉを大きくするに従って$X^{i+1}(h^{i+1})$の平衡状態の値$X(h)$が算出できる。

第６章

付録６.１：アレーディスク装置の適用性

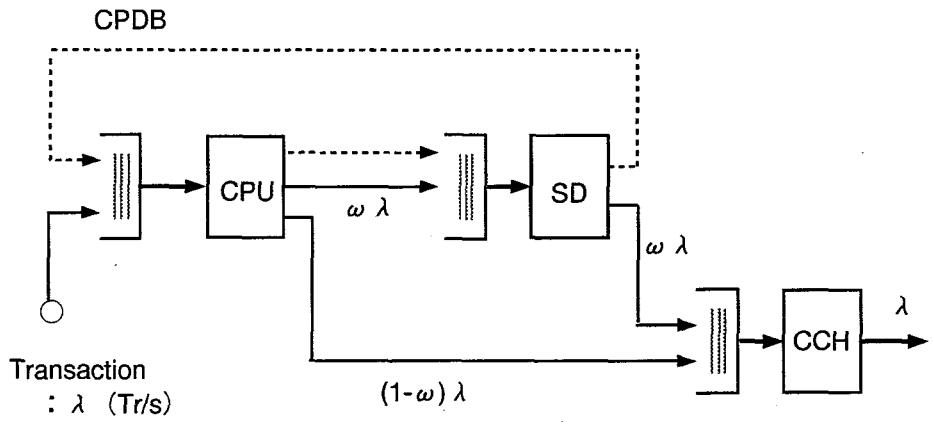
磁気ディスク装置を数台～１０数台使用し、並列に動作可能としたアレーディスク装置がある。最近のアレーディスク装置の単位時間当たりの入出力回数は数１００（回/秒）程度であり、ＬＯＧ取得用としての適用の可能性がある。しかし、数ＧＢの磁気ディスク装置を使用したアレーディスク装置の記憶容量は数１０ＧＢ程度となり、数１００ＭＢ～１ＧＢ程度のデータベースのＣＰＤＢやＬＯＧを取得する記憶装置として使用する場合は、容量面での使用効率が極端に低下し、コスト性能比が悪い。

付録６.２：方式１のレスポンスタイムの評価

ＣＰＤＢはb_{CP}単位にＳＤに書込む。書込みを確実に行うため書込み終了の確認をとる。すべてのＣＰＤＢをT_{CP}内に書込むためには、$T_{CP}\cdot b_{CP}/M$の周期で書込めば良いが、ここではトランザクション処理への影響が最も厳しいケースとして、b_{CP}の書込み終了と同時に次の書込みの要求をＣＰＵキューに登録することとした。シミュレーション評価モデルを図Ａ・６.２.１に示す。トランザクションはポアソン到着とし、平均発生頻度をλとした。b_{CP}とレスポンスタイムの関係を図Ａ・６.２.２に示す。図Ａ・６.２.２よりb_{CP}が小さい場合にはＳＤでの待ちがほとんど問題とならないが、b_{CP}が１００ＫＢより大きくなるに従ってＳＤでの待ちが急激に増大しレスポンスタイムが悪化する。

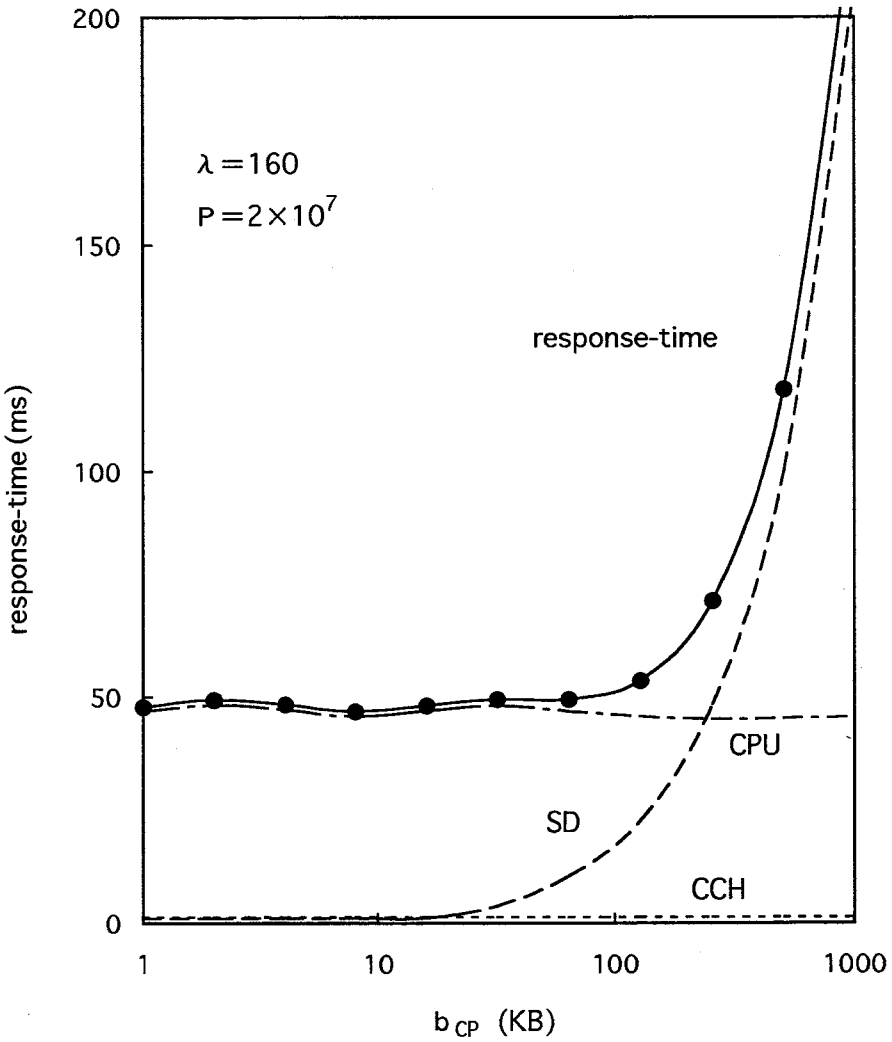
付録６.３：キャッシュメモリ上の最新情報の掃き出し

最近の高性能マイクロプロセッサではキャッシュメモリが使用される。しかも、キャッシュメモリの制御方式として、ほとんどの場合コピーバック方式がとられる。コピーバック方式では、最新のデータはキャッシュメモリ上に存在し、データをキャッシュメモリから追い出すときに初めて主メモリに反映される。従って、電源障害が発生した場合、障害時点にすべてのデータベースが主メモリ上に反映されている保証はない。このため、キャッシュメモリの内容を主メモリに掃き出す必要がある。マイクロプロセッサの書き換えを即座に主メモリに反映するためには、キャッシュ制御方式としてストアスルー方式を採用する必要があるが、ストアスルー方式ではプロセッサと主メモリ間のボトルネックが発生し、マイクロプロセッサの性能を十分に引き出せない。



CPU: central processing unit
SD: semiconductor disk device
CCH: communication control channel
CPDB: check-point database

図A・6.2.1　レスポンスタイムの評価モデル



図A・6.2.2　b_{CP}とレスポンスタイムの関係

第７章

付録７.１：経過時間と不稼働率の関係の算出

本文中の状態遷移の方程式を基に、経過時間と不稼働率の関係を差分方程式を用いて算出した。概要を以下に示す。

ある時間ｔから微小時間δ経過した場合、システムの各状態の確率は次式で近似できる。

$$P_{ij,kl}(t+\delta) \cong P_{ij,kl}(t) + \delta P_{ij,kl}{}'(t) \qquad \text{(A·7.1.1)}$$

式（A·7.1.1）を本文中の式（7.1）～（7.16）に代入し、さらに、駆付けを全く行わないことから、本文中の式（7.24）を適用し次式を得る。

$$\mathbf{P}(t+\delta) = \mathbf{A} \bullet \mathbf{P}(t) \qquad \text{(A·7.1.2)}$$

ここで、**A**、**P**は以下の行列で表せる。

$$\mathbf{A} = \begin{bmatrix} 1-4\lambda\delta & 0 & \cdots & \cdots & 0 & 0 \\ 4\lambda\delta & 1-3\lambda\delta & \cdots & \cdots & 0 & 0 \\ \cdots & \cdots & \cdots & \cdots & \cdots & \cdots \\ \cdots & \cdots & \cdots & \cdots & \cdots & \cdots \\ 0 & 0 & \cdots & \cdots & 1-2\mu\delta & 0 \\ 0 & 0 & \cdots & \cdots & 0 & 1-4\mu\delta \end{bmatrix} \qquad \text{(A·7.1.3)}$$

$$\mathbf{P}(t) = \begin{bmatrix} P_{00,00}(t) \\ P_{00,10}(t) \\ \cdots \\ \cdots \\ P_{20,21}(t) \\ P_{21,21}(t) \end{bmatrix} \qquad \text{(A·7.1.4)}$$

時間ｔからｎδ時間経過後は以下の通りとなる。

$$\mathbf{P}(t+n\delta) = \mathbf{A}^n \bullet \mathbf{P}(t) \qquad (A\cdot 7.1.5)$$

ここで、ｔ＝０とおくと次式となり、ｔ＝０からｎδ時間経過後の各状態の確率が求まる。

$$\mathbf{P}(n\delta) = \mathbf{A}^n \bullet \mathbf{P}(0) \qquad (A\cdot 7.1.6)$$

なお、ｔ＝０のときの$\mathbf{P}(0)$は本文中の式（7.25）、（7.26）で与えられる。

各状態の確率から、ｎδ時間経過後の不稼働率は次式で表せる。

$$P_{SDN}(n\delta) = P_{20,20}(n\delta) + P_{20,21}(n\delta) + P_{21,21}(n\delta) \qquad (A\cdot 7.1.7)$$

δを小さく設定すことにより、精度が高くなる。ここでは、δを１/１０００時間刻みで算出した。これ以上、δを小さくしても結果はほとんど変化なかった。なお、式（A･7.1.5）の数値計算には、Ｍathmaticaを使用した。

略号一覧

ＡＭＣ：ATM connector：ＡＴＭコネクタ

サービス制御ノード（ＳＣＰ）を複数のモジュールで構成する場合、モジュール間をＡＴＭにより接続する結合機構。

ＡＰＬ：application program：アプリケーションプログラム

サービスを実現するための応用プログラム。

ＡＴＭ：asynchronous transfer mode：非同期転送モード

広帯域の伝送／交換技術。本論文ではモジュールと結合機構との間のデータ転送に使用。

ＢＴ：battery：バッテリ

電源断においても、半導体ディスク装置上の情報を不揮発化するために使用。

ＣＣＨ：communication channel：通信チャネル

モジュール間の通信を行うためのチャネル。

ＣＣＭ：communication control module：通信制御モジュール

サービス制御ノード（ＳＣＰ）を機能分散／負荷分散で構成する場合のモジュールの一種であり、共通線信号網を介して交換機（ＳＳＰ）との通信機能を実現するモジュール。

ＣＰ：check point：チェックポイント

ソフトウェアのバグやハードウェア障害により、主メモリ上のデータベースが破壊された場合、障害直前のデータベースを復元するため、ある時間間隔で主メモリ上のデータベースを半導体ディスク装置にコピーするが、このコピーを取得する契機を与える時間のこと。

ＣＰＤＢ：check point database：チエックポイントデータベース

ソフトウェアのバグやハードウェア障害により、主メモリ上のデータベースが破壊された場合、障害直前のデータベースを復元するため、チェックポイント毎に取得したデータベースのこと。

ＣＰＵ：central processing unit：中央演算処理装置

プロセッサを構成する一要素であり、主メモリ上の命令を実行する装置。

ＤＢ：database：データベース

カスタマ毎のデータから構成されるデータの集合。

ＤＢＣ：database copy：データベース　コピー

ソフトウェアのバグやハードウェア障害により、主メモリ上のデータベースが破壊された場合、それを復元するため、他のモジュールの主メモリに格納しておくデータベースのコピー。

ＤＲＡＭ：dynamic random access memory：ダイナミック　ランダムアクセスメモリ

主メモリや半導体ディスク装置で用いるメモリ素子。

ＤＳ：dynamic step：ダイナミックステップ

ある機能を実現するために処理しなければならない命令数のこと。要求される機能を実現するためのプロセッサの負荷を表す尺度として用いる。

ＦＤＤＩ：fiber distributed data interface：エフディディアイ

１００Ｍビット／秒のリング型ＬＡＮの名称。

ＨＤ：hard disk：磁気ディスク装置

ＯＳやＡＰＬのプログラムファイル等を格納する、磁気を用いた記憶装置。

ＨＤＣ：hard disk controller：磁気ディスク制御装置

磁気ディスク装置へのアクセスを制御する装置。

ＩＮ：intelligent network：インテリジェントネットワーク

ネットワーク内にカスタマ毎のデータを保持し、この内容に従って高度な電話サービスを実現する網。

ＩＲＯＳ：interface for realtime operating system：アイロス

ＮＴＴで開発している交換機やサービス制御ノード等で使用しているリアルタイムＯＳ。

ＩＴＵ－Ｔ：international telecommunication union-telecommunication sector：国際電気通信連合標準化部門

電気通信に係わる標準化を行っている国際機関。

ＬＡＮ：local area network；ローカルエリアネットワーク

同一建物内、あるいは同一敷地内などの比較的狭い地域に分散設置されたモジュールなどを結ぶ構内ネットワークシステム。

ＬＯＧ：log：ログ

トランザクションによりデータベースを更新した場合の更新内容。データベースが破壊された場合、それを復元するために用いる。

ＭＭ：main memory：主メモリ

プロセッサを構成する一要素であり、命令やデータを格納する記憶装置。

ＭＵＸ：multiplexing and demultiplexing：多重／分離化部

ＡＭＣを構成する機能ブロックの一種であり、ＳＤＨＩからの信号の多重化や、ＳＤＨＩへの信号の分離化を行う機能ブロック。

ＮＤＭ：network directory module：網ディレクトリモジュール

サービス制御ノード（ＳＣＰ）を機能分散／負荷分散で構成する場合のモジュールの一種であり、カスタマ毎のサービスデータが配備されているＳＣＰを特定するための網ディレクトリを格納し、それに対するアクセスを行うモジュール。

ＯＬＴＰ：on-line transaction processing：オンライントランザクション処理

銀行のオンラインシステムやクレジット照会システムなど、データベースを有するホストシステムと、通信回線で接続された多数の端末とから構成されたシステムで、端末からの問合わせに対してホストシステムでデータベースの更新や参照を行い、結果を端末に返す処理のこと。

ＯＭＭ：operation and maintenance module：オペレーション管理モジュール

サービス制御ノード（ＳＣＰ）を機能分散／負荷分散で構成する場合のモジュールの一種であり、サービス管理ノードやオペレーションノードとの通信機能を実現するモジュール。

ＯＰＳ：operation system：オペレーションノード

インテリジェントネットワークを構成する通信ノードの一種であり、多数のサービス制御ノード（ＳＣＰ）の障害処理や、ソフトウェアの変更に伴うファイル更新処理等を行うノード。

ＰＨＳ：personal handy-phone system：簡易型携帯電話

携帯電話の一種。本論文で対象とするサービス制御ノードにはＰＨＳの位置情報がデータベースとして格納される。

Ｐroc：processor：プロセッサ

ＣＰＵや主メモリ、チャネル等から構成される処理装置のこと。

ＳＣＦ：service control function：サービス制御機能

ＩＴＵ－Ｔで規定されているサービス制御ノードを実現するための機能の一つであり、呼の接続制御等のサービス制御を実現する機能。

ＳＣＭ：service control module：サービス制御モジュール

サービス制御ノード（ＳＣＰ）を機能分散／負荷分散で構成する場合のモジュールの一種であり、サービス制御機能（ＳＣＦ）を実現するモジュール。

ＳＣＰ：service control point：サービス制御ノード

インテリジェントネットワークを構成する通信ノードの一種であり、カスタマ毎のデータベースを保持し、交換機（ＳＳＰ）と協調して、高機能な電話呼の接続処理を行うノード。

ＳＤ：semiconductor disk device：半導体ディスク装置

半導体メモリを用いて構成し、チャネルを経由してアクセスできる記憶装置。機械的な動作を伴わないため、磁気ディスク装置に比べてアクセス時間が短い。

ＳＤＣ：semiconductor disk device controller：半導体ディスク制御装置

半導体ディスク装置へのアクセスを制御する装置。

ＳＤＦ：service data function：サービスデータ機能

ＩＴＵ－Ｔで規定されているサービス制御ノードを実現するための機能の一つであり、呼の接続のために必要なカスタマ毎のサービスデータを管理し、それに対するアクセスを実現する機能

ＳＤＨＩ：synchronous digital hierarchy interface：同期ディジタルハイアラーキインタフェース部

ＡＭＣを構成する機能ブロックの一種で、モジュールとＳＤＨで接続する機能ブロック。

ＳＤＭ：service data module：サービスデータモジュール

サービス制御ノード（ＳＣＰ）を機能分散／負荷分散で構成する場合のモジュールの一種であり、サービスデータ機能（ＳＤＦ）を実現するモジュール。

ＳＩＭ：system interface module：システムインタフェースモジュール

サービス制御ノード（ＳＣＰ）を機能分散／負荷分散で構成する場合のモジュールの一種であり、モジュール間を接続するＡＭＣとＯＭＭを一体化したモジュール。

ＳＬＰ：service logic program：サービス論理プログラム

カスタマが受けるサービス（たとえば、フリーダイヤルの場合、時間帯により接続先の電話機を変更するとか、発信者の位置により接続する電話機を変更する等のサービス）を実現するため、カスタマ毎に作成されたプログラム。ＳＬＰを書き換えることによりサービスのカスタマイズ化が容易に実現できる。

ＳＭＳ：service management system：サービス管理ノード

インテリジェントネットワークを構成する通信ノードの一種であり、カスタマの氏名や住所等のカスタマ情報や、カスタマが受けるサービス条件等のデータベースを管理するノード。実際の呼処理に必要な情報はＳＭＳからサービス制御ノード（ＳＣＰ）にダウンロードされる。

ＳＲＳＷ：self-routing switch：自己ルーチングスイッチ

ＡＭＣを構成する機能ブロックの一種であり、ＡＴＭ信号のルーティングを行う機能ブロック。

ＳＳＰ：service swiching point：交換機

電話呼の接続処理を行うノード。高機能な電話呼（ＩＮ呼）の場合は、ＳＳＰがＩＮ呼であることを検出すると共通線信号網を介してサービス制御ノード（ＳＣＰ）に問い合わせをかけ、ＳＳＰとＳＣＰが協調してＩＮ呼の接続を行う。

ＴＣＡＰ：transaction capabilities application part：トランザクション機能応用部

共通線信号網で、データベースへのアクセスなどを効率的に行うためのプロトコル。

ＴＭＮ：telecommunication management network：通信管理ネットワーク体系

オペレーションノードと、交換機、伝送装置等のネットワークエレメントとを容易に接続するための通信アーキテクチャ。

ＶＣＣＩ：voluntary control council for interference by information technology equipment：情報処理装置等電波障害自主規制協議会

電子装置からの妨害電波の自主規制を行っている協議会。

ＶＰＮ：virtual private network：仮想私設網

インテリジェントネットワークを用いた電話サービスの一種で、加入電話網をあたかも社内の内線電話のように利用できるようにしたサービス。

発表論文一覧　（下線は関連発表論文を示す）

A．論文誌

（1）福村好美，平野正則，塩澤恒道，“バス結合型マルチプロセッサのキャッシュメモリ構成方式,” 信学論(D-I)，vol.J74-D-I，no.10，pp.721-728，Oct. 1991.

（2）木ノ内康夫，桜井紀彦，平野正則，塩澤恒道，“長短データアクセス混在時のストライプアレーディスクのスループット解析,” 信学論(D-I)，vol.J76-D-I，no.8，pp.417-428，Aug. 1992.

（3）平野正則，鈴木孝至，塩澤恒道，芳西　崇，木ノ内康夫，“分散処理による高度ＩＮ用サービス制御ノードの構成,” 信学論(B-I)，vol.J79-B-I，no.8，pp.539-550，Aug. 1996.

（4）平野正則，塩澤恒道，木ノ内康夫，鈴木孝至，“高度ＩＮの分散処理におけるデータ配置による負荷の偏り,” 信学論(B-I)，vol.J80-B-I，no.2，pp.87-97，Feb. 1997.

（5）平野正則，山根道広，山崎幹夫，木ノ内康夫，林　誠治，“高度ＩＮ用サービス制御ノードにおけるメモリデータベースのリカバリ方式,” 信学論(B-I)，vol.J80-B-I，no.8，pp.596-608，Aug. 1997.

（6）平野正則，櫻井秀紀，今川仁，木ノ内康夫，“分散処理による高度ＩＮ用サービス制御ノードのモジュール間結合方式,” 信学論(B-I)，vol.J81-B-I，no.8，pp.519-530，Aug. 1998.

（7）平野正則，山根道広，小林正光，木ノ内康夫，“相互にバックアップされたデータベースを持つ二重化ノードにおける駆付け保守の評価,” 信学論(B-I)，vol.J81-B-I，no.10，pp.***-***，Oct. 1998.

B．国際会議

（1）M. Hirano，Y. Kinouchi，and T. Suzuki，“Distributed Control Node Architecture in the Advanced Intelligent Network,” ISS '95，April 1995.

（２）M. Hirano, T. Shiozawa, Y. Kinouchi, and T. Suzuki, “Large-Scale Distributed Control Node in the Advanced Intelligent Network,” APCC '95, June 1995.

（３）H. Sakurai, M. Ito, M. Hirano, and H.Imagawa, “Inter-module Connections in Distributed-processing Service Control Points in the Advanced Intelligent Network and Testing of Associated Hardware,” ICIN '96, Nov. 1996.

Ｃ．ＮＴＴ社内論文誌

（１）多嶋清次郎，松本博幸，平野正則，小濱晴雄，“ＤＩＰＳ　ＶＬＳＩプロセッサ本体系の構成，” 研実報，vol.33，no.1，pp.17-30，1984.

（２）S. Tajima, H. Matsumoto, and M. Hirano, “Central Processing Subsystem for DIPS VLSI Processor,” Rev. of the ECL, vol.32, no.4, pp.700-708, 1984.

（３）平野正則，山口利和，東海林敏夫，北村士守，“ＤＩＰＳ　Ｖシリーズ用汎用インタフェース制御装置の構成，” 研実報，vol.36，no.6，pp.761-770，1987.

（４）矢沢良一，平野正則，山口利和，岡田靖史，“ＤＩＰＳ－Ｖ３０Ｅのハードウェア構成，” 研実報，vol.37，no.9，pp.523-532，1988.

（５）矢沢良一，平野正則，小濱晴雄，丸山正人，“ＤＩＰＳ－Ｖ３０ＥＸのハードウェア構成，” NTT R&D, vol.38, no.8, pp.885-894, 1989.

Ｄ．その他の機関誌

（１）平野正則，矢沢良一，園田雅文，“ＩＮＳネットへの直接接続を実現した分散プロセッサ，” ＮＴＴ技術ジャーナル，no.8，pp.55-57，1989.

（２）脇村慶明，岡田勝行，小町谷忠芳，平野正則，“高性能システムバスの構成技術を開発，” ＮＴＴ技術ジャーナル，no.6，pp.81-84，1990.

Ｅ．全国大会

（１）桑原敏，平野正則，“マイクロプログラム制御のチャネルにおける多重処理の影響について，” 昭和５２年度電子通信学会総合全国大会，1381，1977.

（2）小原和博，平野正則，多嶋清次郎，“マルチプロセッサシステムにおけるバス制御方式の検討,” 昭和５５年度情報処理学会第２１回全国大会，4G-6，1980.

（3）平野正則，“アドレス変換制御部のＬＳＩ化に関する一考察,” 情報処理学会第２２回（昭和５６年前期）全国大会，7J-5，1981.

（4）平野正則，小原永，“ＶＬＳＩ化論理設計法に関する一考察,” 電子通信学会情報・システム部門全国大会，551，1981.

（5）平野正則,“密結合マルチプロセッサのメモリアクセス方式に関する一考察,” 情報処理学会第３４回（昭和６２年前期）全国大会，5Q-5，1987.

（6）森啓，平野正則,“スプリット制御バスに適したキャッシュ一致制御方式,” 電子情報通信学会春季全国大会，D-120，1990.

（7）塩澤恒道，平野正則，“密結合マルチプロセッサにおけるキャッシュ制御方式,” 情報処理学会第４０回（平成２年前期）全国大会，4L-1，1990.

（8）平野正則，“密結合マルチプロセッサにおける二階層キャッシュ制御方式,” 電子情報通信学会秋季全国大会，D-80，1990.

（9）平野正則，塩澤恒道，福村好美，“バス結合型マルチプロセッサ構成に関する一考察,” 電子情報通信学会春季全国大会，D-141，1991.

(10) 平野正則，塩澤恒道，木ノ内康夫，中村篤，井上潮，“分散データベースシステムのコストパフォーマンス評価,” 電子情報通信学会春季大会，D104，1992.

(11) 塩澤恒道，平野正則，“排他制御方式に関する一考察,” 電子情報通信学会春季大会，D-123，1992.

(12) 塩澤恒道，平野正則，“コピーバックキャッシュメモリを用いたメモリリカバリ方式に関する一考察,” 電子情報通信学会秋季大会，D-56，1992.

(13) 塩澤恒道，平野正則，“メモリリカバリ機能を有するキャッシュメモリ適用法の検討,” 電子情報通信学会秋季大会，D-74，1993.

(14) 木ノ内康夫，平野正則，塩澤恒道，“連続運転の容易化に向けた時空間多相プロセッサの適用について,” 電子情報通信学会春季大会，SD-6-2，1994.

(15) 山根道広，平野正則，木ノ内康夫，“センタ間バックアップを行う高信頼化システムの平均修理時間,” 電子情報通信学会春季大会，A-389，1994.

(16) 櫻井秀紀，塩澤恒道，平野正則，今川仁，“ＡＴＭ結合機構を用いた分散処理システムの試験方式,” 電子情報通信学会ソサイエティ大会，B-406，1995.

(17) 林誠治，平野正則，石橋宏純，小林伸幸，白石正裕，“高度ＩＮ用データベースのリカバリ方式に関する一考察,” 電子情報通信学会総合大会，B-658，1996.

(18) 斉藤秀一，大光明直孝，櫻井秀紀，伊藤守夫，平野正則，“分散処理による高度ＩＮ用サービス制御ノードのモジュール間結合方式に関する一考察,” 電子情報通信学会総合大会，B-695，1996.

(19) 富上幸成，平野正則，大光明直孝，吉見正信，木ノ内康夫，“高度ＩＮにおける分散データベース処理の負荷平準化に関する一考察,” 電子情報通信学会ソサイエティ大会，B-623，1996.

(20) 斉藤秀一，伊藤守夫，平野正則，櫻井秀紀，林誠治，“高度ＩＮ用半導体ファイル装置のバッテリバックアップ方法,” 電子情報通信学会総合大会，B-6-118，1997.

(21) 大光明直孝，林誠治，富上幸成，平野正則，“分散構成におけるデータベースのバックアップ方法,” 電子情報通信学会総合大会，D-6-22，1997.

(22) 川瀬克之，秋葉昭浩，大光明直孝，平野正則，“高度ＩＮ用大容量半導体ファイル装置のメモリ試験に関する一検討,” 電子情報通信学会総合大会，B-6-4，1998.

(23) 山根道広，秋葉昭浩，平野正則，“半導体ファイル用バッテリバックアップ電源の高信頼化構成,” 電子情報通信学会総合大会，B-6-66，1998.