分散形データベース管理システム（DEIMS－3）の構成

鈴木 健司　岡田 静夫　伊藤 健治　田中 豪
（日本電信電話公社 横須賀電気通信研究所）

1. はじめに

DEIMS－3[1]*1は電電公社データ通信用のDIPS*2オペレーティングシステムのもとで走行する分散形データベース管理システムである。DEIMS－3は、これまでに開発してきたCODASYL DBTG提案に基づく集中形DBMSであるDEIMS－2をベースに分散形データベース機能の拡充を図った。

DEIMS－3の目的は、地理的に分散配置されたデータを統合的なデータベースとして構成し集中制御を可能とすることである。その設計目標は次の通りである。

(1) 統一された設計（top-down design）あるいは既存するデータベースの統合設計（bottom-up design）のいずれの形態をも満たす全体スキーマ構成の実現
(2) アプリケーション・インタフェースに対する分散不可視性の実現
(3) ノード間の従属性の排除による相互独立性の実現
(4) 地理的な分散とセンタ内の機能分散、即ち、グローバル・ネットワークとバックエンド・アーキテクチャに対する統一的アーキテクチャの実現

このような設計目標の実現にあたって、最も重点を置く適用システムは、容量が数100GBで処理件数が秒当り約100件という大規模で高トラヒックな実時間システムである。

本稿では分散形データベース機能の拡充にあたり新たに生じる問題として、スキーマ構成、分散単位と配置方法、分散問合せ処理に適したアクセス方法、分散構成におけるインテグリティの保証に対する上記適用における解決方法と実現について述べる。

2. 分散形DBMSの設計

2・1 スキーマ構成

分散形データベースの設計においては、システム全体のデータ定義情報・配置情報をどのように統一的に設定するか、即ち全体スキーマ構成の設定課題とDBMS間にどのような統一的な通信レベルを設定するかという主要課題がある。[2][3]

スキーマ構成については、DEIMS－2において論理・格納・物理および仮想スキーマの4階層スキーマを実現している。[4] 分散形DBMSのスキーマとして解決すべき問題点は以下の2点である。

(1) ネットワークに対する共通データビューの設定
各ノードに存在する種々のローカルなデータベースは、ネットワークに対するビューとして、異種性の排除と部分的なビューの表現を必要とする。

(2) 統合化
ローカルなデータベースは、ネットワークにおいて論理的に統合されたビュ

*1 DEIMS－3 : DEndenkosha Information Management System version-3
*2 DIPS : Dendenkosha Information Processing System

ーとしての表現を必要とする。

DEIMS－3では以上の解決として、以下に示す4つのスキーマ階層を設定した。また、それらのデータモデル選定の条件は大容量で高トラヒックな実時間データベースシステムを構築できるようにCODASYLモデルを基調とした。これらのスキーマ構成を図1に示す。

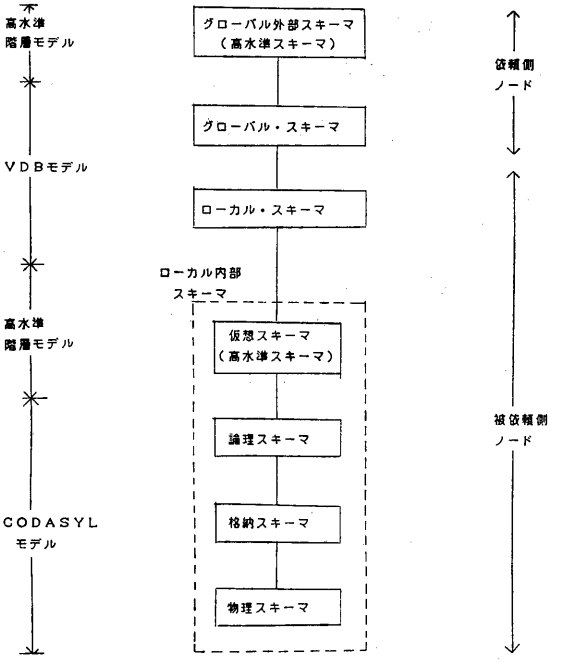


図1　DEIMS－3分散スキーマ構成

(1) ローカル内部スキーマ

ローカル内部スキーマは各ローカルノードに存在するデータビューの記述である。DEIMS－2のCODASYLモデルによる論理・格納・物理・仮想スキーマが対応する。

(2) ローカル・スキーマ

ローカル・スキーマは各ノードに分散配置されたデータベースのネットワークに対するデータビューを記述したものである。その目的は各ノードの異ったDBMSによるデータモデルを均質化することにある。そのデータモデルとして、図2に示すデータ通信網アーキテクチャ（DCNA[*3]）のデータモデル（∇DB）を採用する。これは∇DBが階層モデル、ネットワークモデル，関係モデルを対象とし、各モデルのデータ表現要素の全てを包含し、通信回数、通信量を最小化する高水準な操作コマンドを持つことによる。[5]

(3) グローバル・スキーマ

グローバル・スキーマは複数のノードに分散して配置された複数のデータベースに対する統合的なビューの記述である。そのデータモデルとして∇DBモデルを使用する。グローバル・スキーマは、各ノード上に配置されたデータベースの論理的なデータ構造記述情報とそのデータ構造で示されるデータがどのノードに配置されているかを示す分散配置情報を持つ。

(4) グローバル外部スキーマ

グローバル外部スキーマは、グローバル・スキーマからアプリケーションに適合するように、必要なデータを抽出したデータビューで記述したものである。本データビューの条件は以下の通りである。

(a) 定形処理用のアプリケーションに適した外部ビューであること
(b) 通信上およびローカルノード内の操作言語との変換効率が良いこと
(c) 通信効率の良いこと

これらの条件に対してDEIMS－3では次のデータモデルを設定した。

① オンライン定形業務において一般的に使用されていて処理効率の良い階層モデルを、CODASYLモデルで表現される木構造に対応させ、これをデータビューとするスキーマを新設する。

---

*3　DCNA：Data Communication Network Architecture

② 上記スキーマのデータビューに適した操作命令を、2・3節において後述する高水準操作命令（HDML）として提供する。[10] これにより本スキーマを高水準スキーマ、このデータモデルを高水準階層モデルという。

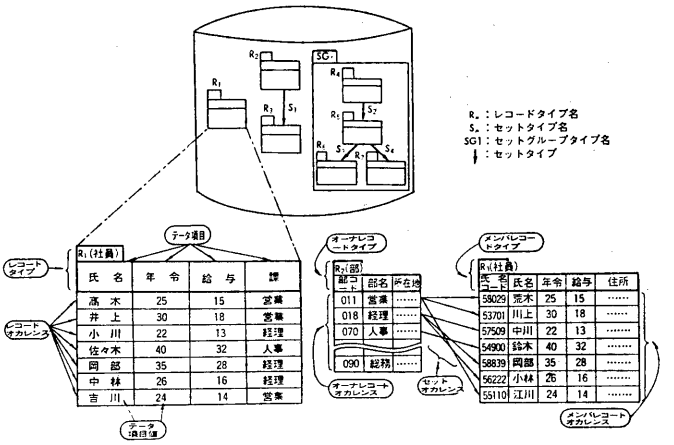


図2　DCNAのデータベース概要

以上がDEIMS－3の分散スキーマ構成とデータモデルの基本設計である。この実現にあたって、第1ステップではDEIMS－3間の同種の分散形DBMSに限定しているため、異種性を排除するためのローカル・スキーマは不要とする。このとき、ローカル・スキーマとして論理スキーマを当てるとローカル内部スキーマにおいて、通信上の高水準な操作コマンドをDMLに変換する必要があり、変換効率上の問題がある。そこでHDMLのまま論理スキーマで処理できるようにDMLと同一レベルでHDMLをサポートした。これにより、ローカル・スキーマとして高水準スキーマを使用する。

2・2　分散配置

分散形データベースの配置問題として、分散単位と配置情報の管理方法がある。

まず、分散単位としてDEIMS－3では外部ビューとして階層構造を設定したことにより、この階層構造に着目し、各ノードのローカルデータベースに対する処理要求が該当ノードで完結できるような階層関係とオカレンスの配置を考慮する。これらよりDEIMS－3では、VDBモデルの構成要素であるセットグループ（SG）タイプを論理的な分散単位とする。SGタイプは階層関係にある複数レコードタイプの集まりである。このSGタイプは複数のノードに重複配置でき、物理的にはデータベースの格納単位であるエリアに複数に対応づけられる。

また、DEIMS－3の第1ステップではデータの重複配置を考慮しないことと処理が該当ノードで完結することを考慮し、SGタイプの実現値であるSGオカレンスは1つのエリアに閉じて格納することにし、1つのエリアは唯一のノードに配置する。即ち、物理的な分散単位はSGオカレンスとエリアである。以上の分散単位の概念を図3に示す。

このような分散配置情報の記述は、エリアあるいはSGオカレンス対応に一意に存在するエントリキーをノード識別のための分散情報として利用する。グローバル・スキーマの分散配置情報の記述項の一般形式を図4に示す。

これによりAPは、分散キー（エントリキー）あるいはエリアを陽に指定することによりノードはDBMSにより識別されるので、データの分散配置に気を配ることなく、SGに基づく処理のみが実施できる。

2・3　分散問合せ処理

DMLを分散形データベースに適用した場合、APとDBMS間のインタフェースは単一レコードであり、APとDBMS間のインタラクション回数が多くなり、通信効率が問題となる。このため次のような高水準インタフェースを設定した。

(1) 複数レコードインタフェース

1つのSGオカレンス全体を操作の基本とし、一度の操作でその構成要素であるレコードオカレンスを一括して扱うことができるように、高水準はレコード選択式を設定した。この選択式はDMLのようなナビゲーションを不要とし、複数のレコードオカレンスを選択できる高水準はレコード選択式に拡張し、操作を容易にしたものである。またAPとDBMSのレコードの授受域であるレコード作業域(UWA)を1レコードタイプ当り任意数のレコードオカレンスを授受できるようにユーザの指定を拡張し、複数レコードインタフェースの操作効率を高めた。

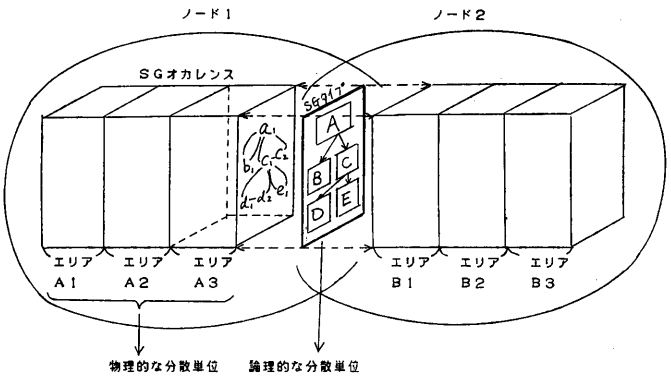


図3　分散単位の概念

《分散記述項》
〈ローカル・スキーマ副記述項〉
LOCAL-SCHEMA NAME IS ローカル・スキーマ名-1
　ON ノード識別子
　;AREA IS {エリア名-1 | FROM エリア名-2 TO エリア名-3} ....

〈レコード副記述項〉
RECORD IS レコード名-1
　;DISTRIBUTION-KEY IS データベース一意名-1
　{(WHERE VALUE IS
　　{EQUAL TO | LESS THAN [OR EQUAL TO] | GREATER THAN [OR EQUAL TO]} 定数-1
　　FROM 定数-2 TO 定数-3
　　THEN IN ローカル・スキーマ名-1 [OF ノード識別子-1]}...
　THEN IN ローカル・スキーマ名-2 [OF ノード識別子-2]}

図4　分散配置情報の一般形式

この機能により、条件に合うデータのみを一度の操作で一括して返却することになるので、DMLより通信効率は向上する。さらに、通信効率を高めるために以下の機能を実現している。

(2) 複合化機能

幾つかの操作命令をまとめて処理させ、実行結果も最終的な処理結果を提示することが可能なHDMLを複合化した実行単位の概念を設け、指定できることとした。

(3) 並列処理

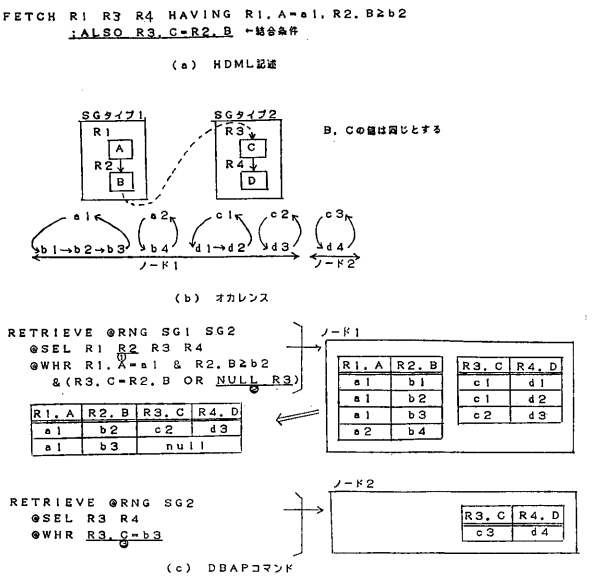
HDMLが一度に複数のSGオカレンスを処理対象とする場合は、操作の対象となるノードが複数になる場合がある。このため、DBMSは問合せを分析した結果、最適化処理として複数のノードへ並行して処理依頼を独立して行える場合は並行処理を行う[8]。

(4) 結合演算における最適化

ノード間に渡る結合演算における通信効率の向上について示す。

本方式は、まず結合側ノード（図5に示す結合条件式の右辺のノード）に対して結合処理を依頼する。結合側ノードでは、結合が完了しなかったものも含めた実行結果をAPの存在する依頼側ノードに返却する。依頼側ノードでは結合が未定のものについて再度条件式を作成し、被結合側ノード（図5に示す結合条件式の左辺のノード）に結合依頼を行う方式[8][9]である。この処理概要を図5に示す。このようなDBMSの問合せ分割処理においては次のような操作を必要とする。

① 結合未定のレコードを転送するための射影条件の追加
② 結合未定のオカレンスを選択するためのnull条件の追加
③ 被結合側ノードへ結合を依頼するために、結合側ノードからの未結合オカレンスから結合キー値を取り出して条件式に設定する処理



（注1）@RNGは操作対象指定，@SELは射影条件の指定，@WHRはレコード選択条件の指定

（注2）①～③は問合せ分割処理のために追加された条件を示す（本文参照）。

図5　ノード間に渡る結合演算の操作例

本方式の実用性は、定形処理の実時間システムではデータ配置に局所性が考慮されており、通常の結合演算の結果は1ノード内で完結する可能性が高く、十分である。

これまでに記述したHDMLを用いたトランザクション処理は、3種類の検索・更新処理においてDMLを用いた処理に比較して、通信回数において0.05～0.2倍、AP－DBMS間のインタラクション回数において0.1～0.2倍、ダイナミックステップ数において0.4～0.8倍程度の性能を得ることができた。特に通信回数は削減効果が大きく、転送するデータ量が増加する程効果が高くなってきている。[6] これにより、分散環境に適した高水準操作言語の実現が可能になったと評価する。

2.4 分散アクセス制御

分散形データベースのインテグリティの保証としての、同一データベースに対する複数トランザクションのアクセス競合とノード間に渡る更新の同期制御について、DEIMS－3での対処について述べる。

(1) アクセス競合制御

DEIMSの集中形データベースの競合制御では、資源競合が発生したときにデッドロックを千エックレ、ウェイトもしくはエラーリターンするLOCK／UNLOCK方式を実現している。

分散形DBMSにおいては、さらにノード間に渡るグローバル・デッドロックの検出を行う必要がある。この対処としてLOCK／UNLOCK方式を採

用すると、他ノードでの資源占有情報を収集しデッドロックを検出する必要がある。このため通信のオーバヘッド，処理ロジックの複雑化が伴う。そこでロジックが簡単で各ノードが単独で処理できオーバヘッドの少い時間監視方式を導入した。この時間監視方式について次に述べる。

データベース資源を要求し、その資源が他のトランザクションにより占有されている場合、ノード内のデッドロック発生の有無の確認を上記方式で実施し、トランザクションをウェイトさせる場合にはウェイトさせる前に時間監視を設定してウェイトさせる。そして一定時間を経過しても応答がない場合には、グローバルなデッドロックが発生したものとみなし、ロールバックさせる方式である。

(2) 更新同期制御

更新同期制御は、集中形データベース制御で確立している救済単位（ＲＵ[*4]）の概念を、ＡＰが存在する依頼側ノードの救済単位をグローバル救済単位（ＧＲＵ）、被依頼側ノードの救済単位をローカル救済単位（ＬＲＵ）と拡張し、２フェーズ・コミットメント方式を導入する。２フェーズ・コミットメントの概要を以下に示す（図６参照）。

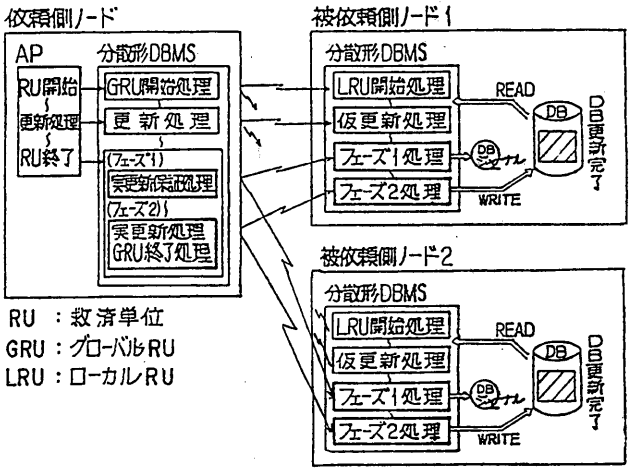


図６　２フェーズ・コミットメントの概要

〔第１フェーズ〕

① 依頼側ノードは、更新を行った全てのノードに対して更新保証を要求し、被依頼側ノードの応答を確認する。

被依頼側ノードの応答が全部肯定応答ならば第２フェーズへ、否定応答を含んでいればＡＰへＧＲＵの正常終了不可の通知をする（ＡＰは次に述べる無効終了指示を行う）。

② 被依頼側ノードではＬＲＵ内でのデータベース更新の保証処理を行い、結果を依頼側ノードに通知する。

〔第２フェーズ〕

① 依頼側ノードは、ＬＲＵの終了を要求し、全ての被依頼側ノードからの応答を受けてＧＲＵを正常終了させる。

② 被依頼側ノードではＬＲＵを正常終了させ、依頼側ノードへ肯定応答を返却する。

〔無効終了処理〕

① 依頼側ノードは、ＬＲＵのロールバック処理を要求し、被依頼側の応答を受けるとロールバック処理正常終了の応答をする。

---

*4 ＲＵ：Rescue Unit；ＡＰの実行とデータベース処理（データベースの更新処理および復元処理等）との同期をとる基本単位

② 被依頼側ノードではロールバック処理を行い、依頼側ノードへ肯定応答を返却する。

具体的な処理は次のようになる。更新保証処理は更新後情報をジャーナル（ＪＮＬ）に取得する。ＬＲＵの正常終了処理は、救済単位正常終了ＪＮＬを取得し、データベースの実更新を実施し、実更新結果のＪＮＬを取得する。ロールバック処理としては、救済単位異常終了ＪＮＬを取得し、ロールバック（Ｉ／Ｏバッファの内容を破棄する）を行い、実更新異常ＪＮＬを取得する。以上の２フェーズ・コミットメントにより更新同期を保証している。[7] これは通信上、ＤＣＮＡのデータベース・アクセス・プロトコル（ＤＢＡＰ）のＳＡＶＥＭＡＲＫ，ＳＡＶＥＥＮＤあるいはＲＯＬＬＢＡＣＫコマンドにより実現されている。

また、分散形データベースにおいては、あるノードがプロセッサ障害や通信系障害により他ノードと切断されることがある。このような障害に対する復旧処理は、自ノードのＪＮＬ情報だけでは不可能な場合がある。例えば、更新保証処理後であっても実更新処理が完了する前にプロセッサ障害が生じた場合、被依頼側ノードのＪＮＬ情報には更新保証情報があるだけでＬＲＵの終了情報が存在しないため、該ノードだけでは該ＬＲＵの結果をデータベースに反映させてよいか判定できない。この場合、ＧＲＵの情報を得ることによって該ＬＲＵの結果をデータベースに反映させる必要がある。

このようにＬＲＵの状態はＧＲＵの状態に一致させねばならないため、ＧＲＵとそれに対応するＬＲＵの状態を管理するために必要な情報を取得する機能をサポートしている。具体的には、ＧＲＵの中で救済単位開始処理依頼時、全ノードの更新保証処理正常時（更新系の場合）、救済単位正常終了時（参照系の場合）および救済単位異常終了時にステータスＪＮＬを取得可能としている。

以上の処理により、分散形データベースのインテグリティの保証を可能としている。

3. ソフトウェア構成

3・1 モジュール構成

データベースを実行時に制御するデータベース・モニタ・プログラム（ＤＢＭ）は、ローカルＤＢＭ（ＬＤＢＭ）とグローバルＤＢＭ（ＧＤＢＭ）から成っている（図７参照）。

ＬＤＢＭは集中形データベースの実行制御を司るモニタで、ＣＯＤＡＳＹＬモデルに基づくＤＭＬを実行するＤＭＬプロセッサ（ＤＰＳ）、高水準階層モデルに基づくＨＤＭＬを実行するＨＤＭＬプロセッサ（ＨＰＳ）およびデータベースへのアクセスを実行する共通プロセッサ（ＣＤＢＭ）から成る。

ＧＤＢＭは分散配置されたデータベースの操作に関するノード間の実行制御を司るモニタで、次の３つのプロセッサより成る。他ノードへのデータベースに関する操作の実行制御を行う分散アクセス制御プロセッサ（ＤＣＳ）、依頼されたトランザクションを代行処理する分散代行処理プロセッサ（ＤＳＳ）および分散形データベースシステムの状態を管理する分散管理プロセッサ（ＤＭＳ）である。

また、ＤＥＩＭＳ－３の通信処理は、複合形分散システム管理（ＣＤＳＭ）を利用している。ＣＤＳＭは、結合形態として チャネル結合と回線結合を対象とし、利用者に上記違いを仮想化して提供するとともにプロセッサ間の通信路が複数個

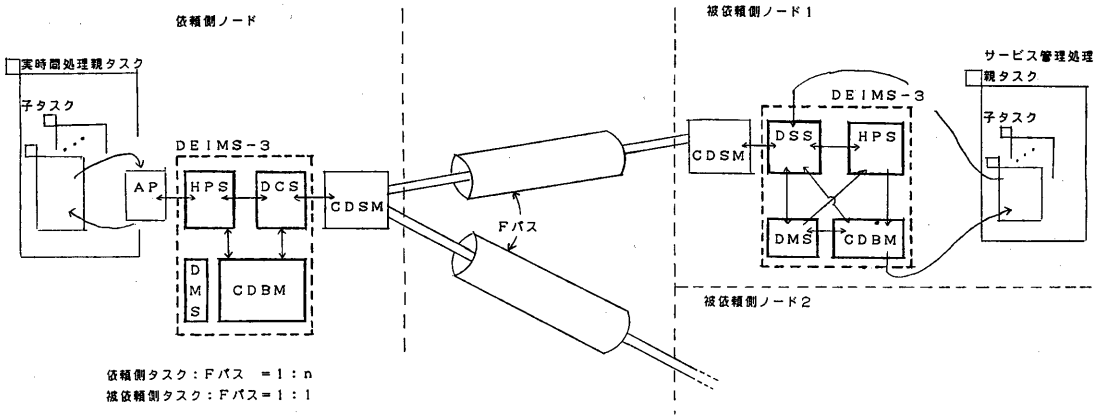


図7　DEIMS-3の位置付け

存在する場合、どの通信路を使っているか意識させず、通信路障害のときは自動的に別通信路にパスを張替える機能を有している。

3・2　処理概要

実時間処理におけるDEIMS-3の位置付けを図7に示す。次にDEIMS-3の初期設定，通常時処理の概要を示す。

(1) 初期設定処理における通信路の設定

DEIMS-3は依頼側ノードではAPが走行するタスク上で走行し、被依頼側ノードではAP処理に対応する代行処理が走行するタスク上で走行する。両ノードのタスク間の対応は、DCNAで規定する情報処理フィールド間の論理的な通信路（Fパス）の設定においてなされる。

Fパスの設定は、データベースシステム作成者側（AS側）で、CDSMのFパス開設機能を利用してノード間において必要とする数だけシステム立上げ時に開設しておく。タスク間のFパスの確定は、データベース処理を実行可能な状態にするATTACH命令において一意に対応付けられる。具体的には、障害時の処理を柔軟に対処可能なように、AS側が一括管理して開設しているFパスを被依頼側ノードのタスク生成時に一意に割付けておく。そして依頼側ノードでATTACH命令発行時、被依頼側ノードのタスクと対応付けられているFパス識別子を、付加ルーチン・インタフェースによりAS側から授受することにより確定する。依頼側ノードと1つの被依頼側ノードのタスク間の関係は1対1であるが、依頼側ノードから複数の被依頼側ノードにFパスを設定可能である。

Fパスの確定契機はATTACH命令時以外に、障害中のFパスに対する代替Fパス確定契機がある。

(2) 処理の流れ

HDML発行後の分散問合せ処理の流れを検索命令（FETCH）を例に以下に示す。この処理概要を図8に示す。

① APがHDML命令（FETCH）を発行

② HPSは高水準スキーマを参照し、分散環境での走行ならばDCSにアク

セスを依頼

③ ＤＣＳはＨＤＭＬの情報とグローバルスキーマの分散配置情報を基にアクセスすべきノードを決定し、依頼コマンド（ＲＥＴＲＩＥＶＥ）を作成し、ＣＤＳＭを用いてコマンドを発行

④ 依頼コマンドを受信したＤＳＳは、コマンドを解釈し依頼側ＡＰに代ってＦＥＴＣＨ命令に対応するＨＰＳマクロを発行し、アクセスを依頼

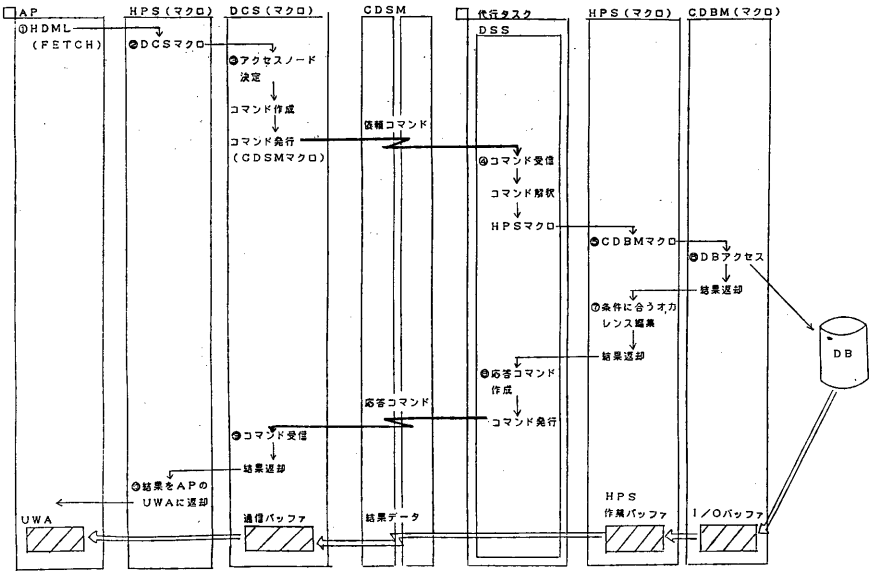


図８　分散同合せ処理

⑤ ＨＰＳはＣＤＢＭに対してデータベースのアクセスを依頼

⑥ ＣＤＢＭはデータベースにアクセスし、結果をＩ／Ｏバッファに返却

⑦ ＨＰＳはＩ／Ｏバッファのレコードから ＨＤＭＬで指定したレコード選択条件に合うレコードを抽出し、ＨＰＳ作業バッファに編集

⑧ ＤＳＳはアクセス結果の情報を返却するための応答コマンド（ＲＥＰＬＹ）を作成し、結果データとともに依頼ノードへ返却

⑨ ＤＣＳは応答コマンドおよび結果データを受信し、ＨＰＳへ通信バッファを介して結果を通知

⑩ ＨＰＳは通信バッファ上のレコードをＡＰの作業域（ＵＷＡ）へ返却

以上の処理の基本は、ＨＤＭＬの条件式によって転送データ量が大きく変動する結果データを効率よく処理することである。このため次の処理を実施している。

(a) 依頼側ノードでは検索依頼に対する結果データが通信バッファに入りきらず残存する場合は、ＡＰに途中までの結果を返却するとともに被依頼側ノードに対して残りのデータ転送依頼を先行して発行する。このため、依頼側ノードの通信バッファは２面としている。これによりデータ授受に要する時間を最適化している。

(b) ＡＰへの結果データの返却はＳＧタイプを基本としている。このため、ＨＰＳの作業バッファへのレコード編集はＳＧオカレンスを基本単位として次のような処理を行い、編集効率を上げている。

(i) n番目のＳＧオカレンスの処理が完了しＨＰＳ作業域に空きが残存する場合、引続き処理を続行するか中断するかをそれまでに編集したＳＧオカレンスの情況により判断している。

(ii) n番目のSGオカレンスを対象とした処理中にHPS作業バッファが満杯となったときには本処理を中断し、処理の完了したSGに対応するレコードオカレンス群を返却する。引続き検索依頼がくると、未返却のレコードオカレンスを該バッファの先頭に再配置した後、中断時点から処理を続行する。

(iii) 1SGオカレンスが本バッファに入らないときは動的拡張を行い、少くとも1SGオカレンスの処理結果を返却する。

4. おわりに

以上述べたDEIMS-3は製造を完了し、適用システムの導入が進められている。

DEIMS-3の今後の課題としては、情報提供形のような検索を主体とし、重複して配置することを必要とする分散形DBMSへの拡張がある。

<参考文献>

[1] Suzuki, K., et.al. : Implementation of a Distributed Data Base Management System for Very Large Real-Time Applications, Proc. of IEEE COMPCON Fall, pp.569-577, 1982

[2] Adiba, M., et.al. : Issues in Distributed Data Base Management Systems, A Technical Overview, Proc. of the 4th VLDB, Berlin, pp. 89-110, 1978

[3] Takizawa, M., and Hamanaka, E., : The Four-schema Concept as the Gross Architecture of Distributed Databases and Heterogeneity Problems, J., Inf., Proc., Vol. 2 No. 3, p.p 134-142, 1979

[4] Toh, T., Kawazu, S. and Suzuki, K. : Multi-level structure of the DBTG Data Model for an Achievement of the Physical Data Independence, Proc. of the 3rd VLDB, Tokyo, p.p. 403-414, 1977

[5] 河津, 柴崎, 南, 大沼 : DCNAのデータベースアクセスプロトコル 通研実報, 30, No. 3, 1981

[6] 戸田, 村田, 田中 : CODASYL DMLの高水準化とその評価, 情処学会第24回全大, 1982

[7] 岸本, 田中, 服部 : 分散形データベースシステムにおける救済制御について, 同上

[8] 村田, 服部, 鈴木 : 分散形データベースのアクセス処理方式について, 同上

[9] 服部, 鈴木, 岸本 : 分散形データベースにおける通信回数削減方式について, 電子通信学会全大, 1981

[10] 田中, 坂本, 村田 : CODASYLデータベース機能の高水準化について, 同上