4G-1 サービスベースシステムにおける分散データの取り扱いについて

深沢 友雄　田中 英彦　元岡 達

(東京大学 工学部)

1. はじめに

ユーザの個人用計算機と、計算機センタ網を接続する場合、各計算機の提供する機能を、自由に組合わせた処理が容易に行なえる事が望まれる。

筆者等は、各計算機を「サービスベース」としてとらえ、計算機間通信におけるハイレベルなモデルを設定する事により、ユーザが、各計算機で独立に提供しているデータやプログラムを、その分散性を意識する事なく利用できる「サービスベースシステム(SBS)」の提案と、実験システムの実装を試みてきた。すなわち、システムは、以下に示す3レベルのビューを実現するサブシステムからなる。

① 内部ビュー(Internal View)
② 概念ビュー(Conceptual View)
③ 外部ビュー(External View)

本稿では、サービスの一つの要素である「データ」をどうモデル化するかを述べ、特に、分散データの取り扱いをどの様に行うかについて述べる。このうち特に、サービスベース機能の核をなす概念ビューの実現方法を中心に述べる。

2. サービスベースにおけるデータのモデル

2.1 データセットとパラメータ

SBSの処理系が或る関数を起動する時、その関数が扱うデータの値を、処理系が直接与える場合と、ファイル名の様に、単に、データの格納場所の名前を与える場合とがある。 前者のデータを「パラメータ」と呼び、後者のデータを「データセット」と呼び、区別する。例えば関数にファイル名を与えて起動する場合、ファイル名は、パラメータであるが、ファイルの内容は、パラメータではない。

2.2 データセットのタイプ

サービスベースでは、データセットのタイプとして、ファイルと、DBの2種類を考える。本研究では、DBにおけるデータのモデルとして、理論的に取り扱い易い関係モデルを用いる。すなわち、データセットとして、ファイルと、Relationの2種類のタイプを取り扱う。

2.3 データセットと関数の関係

Relationを扱う場合は、データとして定義する際に、ユーザがスキーマを記述する。Relationに作用する関数を定義する場合は、その関数のRelationに対するビューも合わせて定義する。処理系は、ユーザが定義したデータのスキーマに基いて、データに関数を適用(Apply)する。

2.4 概念ビュー上でのデータセットの内容の取り扱い

ファイルの内容を、概念ビュー上のデータの値として取り出す基本サービスを用意する事により、データセットの内容への、関数の適用を可能にする。

3. 分散データの取り扱い

3.1 パラメータに関して

変数を用いてパラメータを指定する場合、変数名の管理が必要となる。

原則として、変数名は、各計算機内でのみ有効とし、他計算機内で値が決まる変数は、externalという修飾子を変数名につける事にする。

3.2 データセットに関して

分散しているデータセットをサービスベースの要素として取り扱う場合、幾つかの実現方法が考えられる。すなわち、

1.) 分散DBシステム、或るいは、分散ファイルシステムを、①のレベルで構成し、②のレベルに対しては、統一的に管理されている様に見せる。
2.) ①のレベルでは、DBや、ファイルは各々独立に定義し、②のレベル上で統合化する。

2.)の方法を用いる場合、起動する関数(ロードモジュール)が存在する計算機に、データを転送する必要がある。この時、SBSでは、次の2つの実現方法が考えられる。(図 3.1参照)

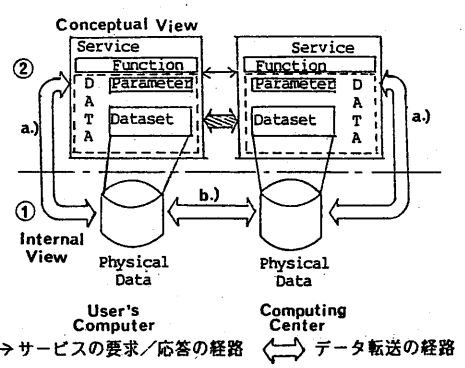

図 3.1 分散データの転送方法

a．）概念ビュー上の通信形式にして送る。
b．）内部ビュー上で、ファイル転送のutility を用意し、基本サービスとして②のレベルに提供する。

3.3 外部ビュー

外部ビュー上では、データの分散を意識せず、任意の計算機上のデータセットに対して、ロードモジュールを作用させる事を可能にする。この為には、Ｆront－Ｅnd側の計算機には、他計算機上のデータに対するスキーマが必要になる。システムは、この情報に基づいて、前節の a.）か b.）のいずれかの方法でファイルを、ロードモジュールの存在する計算機に渡し、ロードモジュールを起動する。

4．実装方法

本研究では、Ｌisp による実験システムを作成してきたので、Ｌisp を使う場合について紹介する。

4.1 通信プリミティブ

②のレベルにおける通信プリミティブとして、
i）（receive　データセット名）
ii）（send　データセット名）
を用意する。各々の機能は、
i）ファイルをopenし、要求先からのデータを、ファイルに入れる。終れば、ファイルをclose する。
ii）eof になるまで、ファイルの内容を、他計算機に送信する。

これにより、図 4.1の様な環境において、②のレベルで、

```
(progn
  (receive File)
  (send file)
  (Function File)) … (＊)
```

を評価すると、（Function　file）…（＊＊）を実現する事ができる。

（receive ・）（send・）は、ファイルの内容を概念ビュー上に取り出す事ができれば②のレベルでも実現できる。（ 3.2の a.）の方法） Ｌisp を用いたシステムでは、ファイルからの１文字入出力の関数（tyi , tyo 等）を用いればよい。 この時、ファイルの内容は、Ｓ－式で受け渡す事になる。効率の点から、実験システムでは、i ）、ii）を①のレベルで実現した。（ 3.2の b.）の方法）

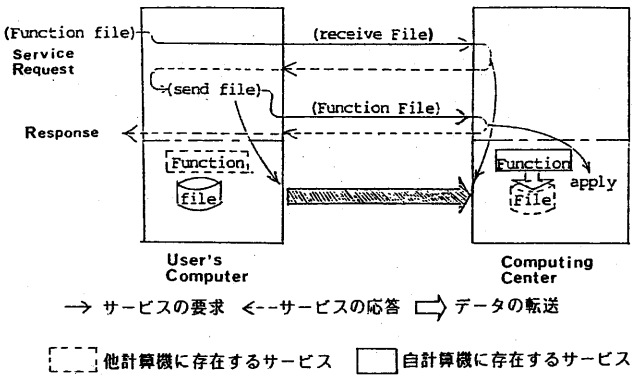

図 4．1 Conceptual View 上で見たデータの転送

4.2 データセットと関数の関係

Ｒelation に関しては、前節の通信プリミティブの他に、
・スキーマの記述
・Ｒelational Ｏperation（ex. 関係代数）
を基本サービスとして用意する。システムは、ユーザの定義したスキーマを管理し、関数のＲelation への適用を行なう。Ｒelatinal Ｏperationの実現も、②のレベル上でＬisp の基本関数を用いて可能であるが、実験システムでは、projection, selection , join, union , difference の関係代数演算を内部ビュー上で用意した。これらを、
（def-service projection selection …）
と定義すれば、基本サービスとして概念ビュー上で利用できる。

他計算機のＤＢにアクセスする場合は、そのＤＢのＤＭＬを基本サービスとして用い、一時データセットを介して、 4.1の方法でデータセットを送受し、自計算機に取り込んでからアクセスする。

5．検　討

前章で述べた様にファイル転送や、関係代数等のサービスも②のレベル上で実現できる。しかし、ファイル転送などの基本的なサービスは、内部ビュー上で実現し、これを基本サービスとして概念ビュー上に登録する事により行うのが自然である。 この時、ＳＢＳでは、ＯＳの提供するロードモジュールを、概念ビュー上で、単に、プリミティブとして登録すれば、サービスとしてすぐ起動できるので、それぞれのサービスは、そのサービスに適した言語で記述する事により、容易に実現できる。

現システムにおいては、まだ外部ビューの提供と、Ｒelation のスキーマの管理を実現していない。
4.1（＊＊）の様な、関数とデータの指定だけでシステムが自動的に、（＊）形式の通信を行なったり、ＤＢやファイルを、ユーザによるデータのビューの定義に基いて、関数が扱える形式のデータに直す機構をサポートする事が、今後の課題である。

＜ 参 考 文 献 ＞

［１］．深沢、田中、元岡、「サービスベースシステムにおけるサービスの構造に関する一考察」、第25回情処全大、昭和57.10.、pp.815-816.