

1U-3

対話型データ転送プログラムの開発時のメモリ制約回避策

清水英雄, 小川将司
富士通㈱ SEテクニカルセンター

1. はじめに

今日パソコン・ソフトの持つ使い勝手の良さや適用業務の範囲の広さにより、オフィス内でのパソコンの利用が益々盛んになっている。

しかもパソコンを利用し大量の情報・データを効果的に処理推進するため、スタンドアロンだけでなくホストのデータをパソコンで処理、加工を行うといった分散処理を目的としたパッケージのニーズが高まってきている。

リレーショナルデータベースの多くはJISやANSI、ISO等によって標準化されたデータベース定義操作言語(SQL)を使っている。

ホストのデータベースでは、コンソールもしくはプログラムからデータベース定義操作言語を使用して、データベースの定義・照会操作を行っているが、この言語は難しく、エンドユーザ向きではない。

そのため、パソコン側流通ソフト(表計算ソフト)から、ホストデータベース

のデータを直接利用できる製品を開発することになった。

2. 開発の前提と問題点

開発を行うに際しては、以下の点が前提となった。

- ① MS-DOSを対象としたプログラム開発
当時、流通ソフトが充実しており、利用者の最も多いMS-DOS上での開発。
- ② 多種ホストアプリケーションの対応
ホストに既存の多種データベースをサポート対象とする。

ところが、この前提条件を満たすためにはいくつかの問題があった。

(1) 厳しいメモリの制限

運用時には、大量のデータが扱えることが必要不可欠であり、多くのワークメモリ領域を確保することが必要であった。しかし、このMS-DOSには、厳しいメモリの制限(FMRシリーズ日本語MS-DOSでは、768KB)がある。その上、オンラインプログラムであるため、通信系ソフトと流通ソフト(表計算ソフト)を当システムとホストコンピュータ

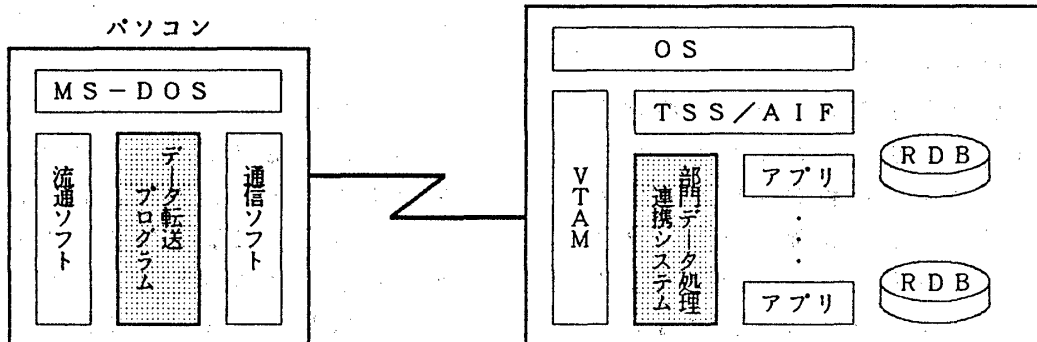


図1 システム構成

同時に動作させなければならず、ワークメモリ領域を除いて、当システムが使用できるメモリを計算すると、約50~40Kバイトとなってしまふ。

(2) データベース操作コマンドの非互換
ユーザの一般的な利用環境では、対象ホストアプリケーションが異なり、データベース操作コマンドにも非互換がある。そのため、ホストアプリケーション毎に各々のコマンドを組み立てるためには、ホストアプリケーション分のコマンド変換処理を行わなければならず、プログラムのサイズが大きくなってしまふ。

3. システム実現の方法

上記の理由により、パソコン側ですべての処理を行うことは難しく、パソコンとホストの双方に連携ソフトを開発した。そして、その間のプロトコル(中間言語等)をパソコン側のプログラムサイズを抑制するように設定した。

(1) 中間言語の決定

ホストアプリケーション毎に異なるSQLライクなコマンド文の非互換を吸収するために、最低限必要な情報を網羅した中間言語を設けた。

パソコン側では、表計算ソフト等の指定内容により、この中間言語の組み立てのみを行い、ホスト側連携ソフトに送信している。

(2) ホスト側連携ソフト

パソコン側から送られてきた中間言語には、ホストアプリケーション単位に対応するコマンド生成情報が網羅されており、この中間言語から各種のホストアプリケーションに対応したコマンドを生成する変換プログラムをホスト側に置いた。そして、各々に対応したデータベースのテーブル名やフィールド属性の取り出し及び条件検索の処理等のホストコマンド(SQL文)の組み立て処理を行うようにした。このようなロジックで行ったこと

により、複数ホストアプリケーションの対応を可能とした。

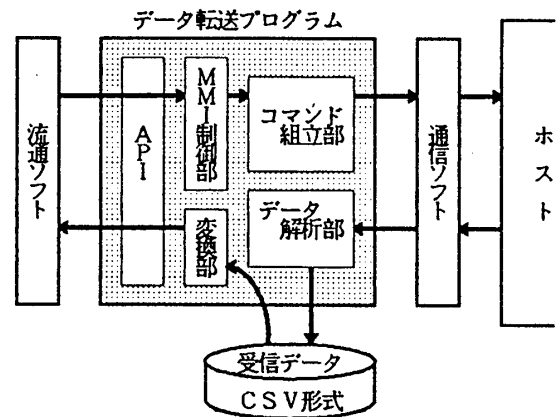


図2 プログラム構造

4. 効果

中間言語を使用しなかった場合は、ホスト側アプリケーション分のコマンド組み立てプログラムが、パソコン側に必要となるが、中間言語の使用により、MS-DOS環境でのメモリの有効活用に対する度合いは、非常に大きくなった。今後、ホストアプリケーションのサポート範囲を広げても、パソコン側に修正を入れることが、避けられる。

今回、開発を行った『データ転送プログラム』は、シングルタスクであるMS-DOS上で、表計算ソフトを起動した状態で、ホストと連携しホストのデータを検索・抽出し、ホストを切断することなくパソコン側の流通表計算ソフトで扱えるといったオンラインプログラムを実現した。

5. おわりに

MS-DOS環境でホスト・パソコン連携を実現することは、市販されている汎用ソフトが多彩な機能を盛り込みサイズが大きくなりつつある今日、メモリの制限上大変、厳しい状況にある。

そのため、今後は、ユーザ層の増加に合わせ、流通ソフトが充実しつつあり、メモリ回避などを気にせずプログラム開発が行えるOS/2などのOS上での開発を検討して行きたいと考えている。