7K-2

機能分散型計算機における データ・サブシステム

吉田浩・田中英彦・元岡達
（東京大学　工学部）

1.はじめに　近年、情報処理の分野において分散処理が脚光を浴びているが、機能分散型計算機の研究もその一つの現われと言える。機能分散型計算機は、特定機能に専用化されたサブシステムを組合わせて構成されるものであるが、その構成法の一つとして、図1のように，システム全体を機能サブシステム、データ・サブシステム，入出力サブシステム、システム管理サブシステムの4つに分割する方法が考えられる。ここでは、このうちのデータ・サブシステムを取上げ、特にその実験システムとしてファームウェアを用いたデータ・サブシステムのファイル・アクセス法μ-VSAMの実装法及び評価・検討について報告する。

2.データ・サブシステムの機能　機能分散型計算機におけるデータ・サブシステムの役割としては、次のようなものがあげられる。

(1) 通常のOSのデータ管理機能を独立させサブシステム化したものとして，システム全体のファイルを管理し、これに対するアクセスの手段を提供する。

(2) より進んだ機能として、データベースの管理や，ファイルに対するアクセス量の多いユーティリティ（ソートやテキスト・エディタなど）を実現する。

このような機能を実現するための基礎として、まずファイル・アクセス法をシステム上に実装する必要がある。このアクセス法は、他サブシステム上のプログラムから直接利用されると共に、(2)で述べたようなデータベース管理システムなどの要求を受けて、実際のファイル・アクセスを行なう。特に機能分散型計算機上では、サブシステム間の通信量をなるべく少なくすることが肝要であり、そのためこのアクセス法のインタフェースの設定には十分な考慮が必要である。

3.μ-VSAMの設計方針　以上のようなデータ・サブシステムの構成法について検討し，さらに評価を行なうために、実験システム上にファイル・アクセス法μ-VSAMを実装したが，その設計に当っては次のような方針をとった。

(1) 同一ファイルに対して種々のアクセス形態（順，乱）が可能である。

(2) ユーザに、ファイルの物理的構造や制約を、極力意識させないようにする。

(3) アクセス法のコマンド体系は、できるだけ高級で、かつ単純なものにする。

(4) サブシステム間のデータ転送量を減らすのに有効な機能（たとえばレコード

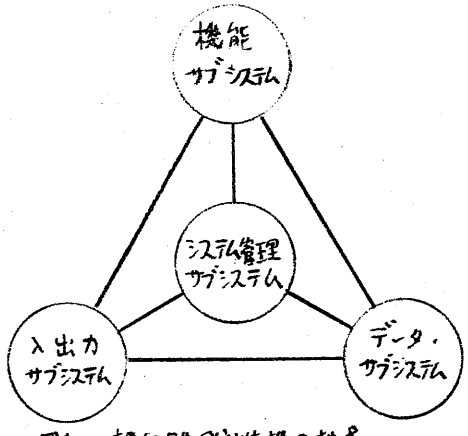

図1.　機能分散型計算機の構成

表1.　μ-VSAMのコマンド

コマンド名	処理内容
create	ファイルの作成（レコード即時入力可）
delete	ファイルの消去
copy	ファイルのコピー，形式変換，再編成
open	ファイルの使用開始処理
close	ファイルの使用終了処理
get	レコードの読出し（フィールド指定，条件指定，複数レコード読出し可）
put	レコードの挿入・更新（フィールド指定可）
erase	レコードの削除（複数レコードの削除可）
pagn	レコードの更新と次レコードの読出し（順アクセスのみ）

中のフィールドの抽出・更新や，条件を満たすレコードのみを読出す機能，1つのコマンドで複数のレコードを処理する機能など）を積極的に導入する。

(5) 関係データベースや，テキスト・エディタなどを上部に構築することを考慮し，さらにこれらの機能の一部（たとえば関係データベースにおける射影や選択の演算など）をアクセス法に吸収して，上部の負担を軽くする。

(6) ファイルの作成，消去，コピー，形式変換，再編成などのファイル保守コマンドをアクセス法レベルで実現し，サブシステム間の通信量低減の一助とする。

(7) 専用プロセッサの構成という観点から，アクセス法をファームウェア化する。

(8) 時分割多重処理を行ない，複数ユーザのコマンドを並行に処理する。

4. μ-VSAMの実装

μ-VSAMは，ポリプロセッサ・システムPPS-1上にファームウェアとして実装された。このシステムには，さらに入出力サブシステムの試作機が接続されており，2つのサブシステムによる分散処理の実験も可能である。

μ-VSAMで取扱うファイルの構造は，通常のOSにおけるVSAMのキー順データ・セットに準拠しており，処理アルゴリズムも大体これに従っているが，会話処理用の機能分散型計算機での使用を念頭に置いて，種々の修正が加えられている。またこの他に，記憶領域の利用効率を考えて，順編成ファイルも取扱っている。表1に，実装したμ-VSAMのコマンドの種類と機能を示す。このマイクロ・プログラムはマイクロ・アセンブリ言語によって記述され，その大きさは約7.2kステップ（21.6kB）で，作成におよそ6人・月の労力を要している。

5. 評価・検討

作成したμ-VSAMにおける主要なコマンドの処理時間の実測結果を表2に示す。また，スループットの上限として，プロセッサで単位時間に処理できるコマンド数を実際のマイクロ・プログラムのステップ数から求めたものが表3であり，この値は汎用中型計算機（FACOM 230-38）の索引順編成ファイルについて測定した結果と同程度のものになっている。

μ-VSAMの実行ステップ数を調べてみると，主記憶中でのデータの移動，チャネルの制御，文字列の比較などの処理時間が比較的多いことがわかり，これらの処理の高速化により，スループットをさらに高めることが可能である。これらに比較すると，VSAMに特有な，レコード挿入の際のコントロール・インタバルの分割法の決定などでは，その複雑さの割には余り処理時間を要していない。

6. おわりに

以上，機能分散型計算機におけるデータ・サブシステムについて述べた。今後の課題としては，μ-VSAMの高速化や機能の拡張，さらにデータベース管理などの高級機能の実現法の検討などが必要と考えられる。

表2. コマンドの処理時間

処理内容	処理時間（msec/コマンド）
ファイルの作成（1000レコード）	4000
乱アクセスのput	127
順アクセスのput（ファイル最後へのレコード追加）	27
乱アクセスのget（ファイル作成直後）	88
乱アクセスのget（1000レコード挿入後）	97
順アクセスのget（ファイル作成直後）	4.0
順アクセスのget（1000レコード挿入後）	3.8

注．レコードは固定長80バイト，キー12バイト，C2%は33%，CA%は27%である。

表3. μ-VSAMのスループット

コマンド	プロセッサ単位処理時間	単位時間内実行回数
乱アクセスget	3.3 msec.	300回
順アクセスget	0.84 msec.	1200回
乱アクセスput	5.8 msec.	170回

注．レコードは固定長で80バイト，キーは8バイト，インデックスの高さは3段でした。また，ファイル中のレコードの削除は行なわれないものと仮定した。