

MVS/ESAにおける常駐DB機能

武藤善八郎

日本アイ・ピー・エム(株) 大型システム・サポート

1988年にIBMから発表されたESA/370(Enterprise Systems Architecture/370)は、ある意味でS/370と同じように大型Computerの使用方法を画期的に変革するものであると同時に今後の大型Systemの発展の土台となるSystemである。このSystemは既に使用可能となり、現在かなりの数のESA Systemが稼働しこのSystemの優秀さが証明されている。このESAの目的はSystemのThroughputを上げ 使いやすいSystemとすることにより、その組織をより戦略的、より活動的にすることにある。この目的に沿った多くの機能を持っているが、ここではこのESAの主機能を用いて達成することのできる常駐DB機能について概説する。

Resident DB Function in MVS/ESA

Zenhachiroh Butoh

System center Large systems support, IBM-Japan, Ltd.
18-24, Tsukiji 7-chome, Chuo-ku, Tokyo 104

The year 1988 was significant in the evolution of IBM's large systems. It was marked by the introduction of the Enterprise Systems Architecture(ESA) and the new Enterprise Systems Architecture/370™(ESA/370™) family of IBM systems and supporting products. ESA represents a significant milestone in the evolution and focus of IBM's large systems. It establishes important architectural and system software foundations that are essential in meeting the emerging requirements of today's large-system customers and crucial to the role evolving for the large systems of the future. This paper reports about one useful example on MVS/ESA™ of 'Resident DB Function' which is a part of 'data in memory' function.

1. はじめに

この報告は、既に商業化されている汎用Computerの上でのシステムの基本思想に密着したOSの一部の主機能の説明とOSに密着した適用プログラムの開発技術にヒントをあたえるものである。しかしこのヒントは応用性のあるものであり、ここで説明するものが唯一無二のものでないばかりか、この通り作成することによりESAでは最高のスループットを得られるとかESAの本来の目的に沿っているということを保証するための報告ではなく、この説明をもとにその適用プログラムに適した各々の解決策を図るための報告であり、この説明はこのために役立つことを希望する。

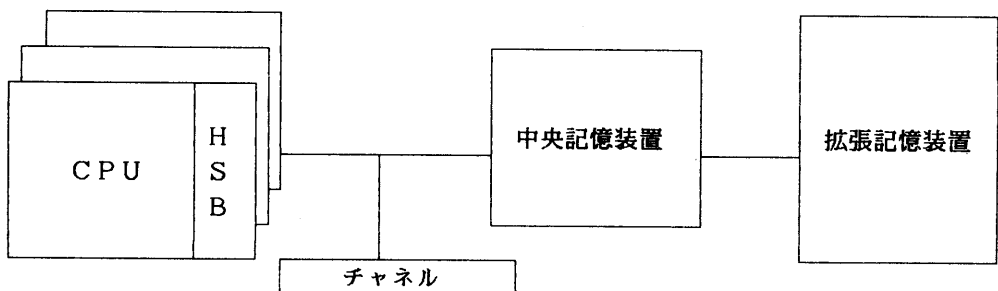
今日のComputer Systemはもはや省力化のためには無く、その組織の機動性、戦略性をたかめるために必要不可欠となっており、そのために必要で重要な要素はComputerの一単位命令のスピードでは無くシステム全体の能力であるスループットである。即ちハード・ウェア能力だけでは無くこのハード・ウェア能力を極限まで引き出すOS、さらにそのOSの援護の下その組織に要求された機能をすばやく処理する適用プログラムが主役である。即ちハード・ウェアの能力は当然の機能であり、ソフト・ウェアの能力がそのシステムを左右し、その組織の生き残りを左右するわけである。

ESAはまさにこのシステムの仕組みを巨大なものにし、より大きなスループットを求められる今日、今後のComputer Systemを支えるためのアーキテクチャーとして登場した。そしてこのESAの下で稼働するOSであるMVS/ESAはその組織のために直接貢献している適用プログラムにこのESA機能を享受させるためのものである。このMVS/ESA機能の一番主要な機能は拡張空間機能を使った実感として掴みづらいくらいの大きな仮想域を提供することにあるがこのなかに空間機能と呼ばれる機能があり、この機能を使って常駐DB機能(IBMの一般用語ではない。)を達成し大きな空間を最大限に有効に利用することにより入出力を極小化し最大限のスループットを得る事が出来る。ここではこの機能に絞ってこの機能の意味及びある一つの効果的な使用例について報告する。

2 MVS/ESAの空間機能

2. 1 MVS/ESAが動くハード・ウェア構成

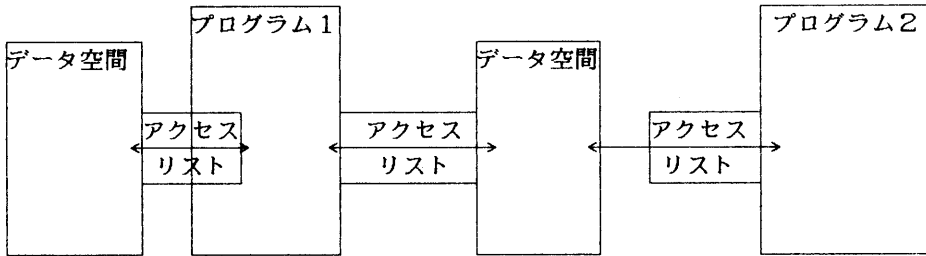
現在MVS/ESAが稼働できるシステムはIBMシステムの4381-9xEとES/3090の各モデルであり以下のような構成が標準的なものになっている。



第一図：プロセッサ内の概略

2.2 データ空間

これまでのMVSと同様に一つのプログラム空間は2ギガ・バイトの大きさであるが、これと同じように一空間2ギガの大きさのデータ(命令のオペランド)だけを入れられる空間をデータ空間と言い以下のごとくこの領域はアクセス・リストと呼ばれるブロックを介してアクセスされ、実際にアクセスするコードはこれまでの命令に何の変更もなくただ各プログラムの先頭でこのデータ空間にベース・レジスタを与えればいいだけである。これにより各プログラムはESAアーキテクチャー上16テラーまでの仮想記憶を同時に使用出来る事となる。このデータ空間はそのプログラム又はある空間のための専用空間と多くの空間で共用できる共用空間とがある。

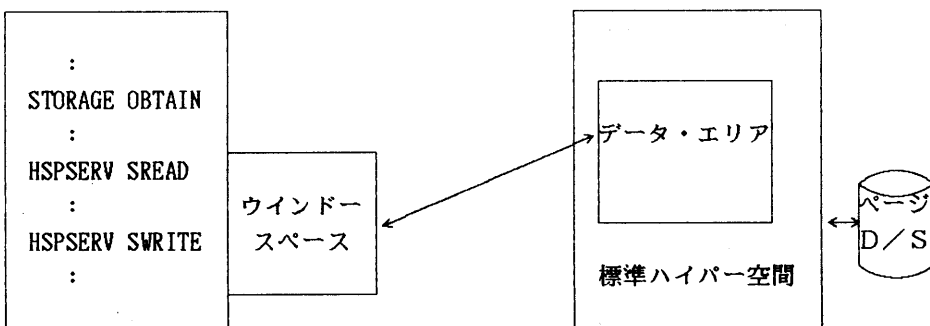


第二図: データ空間

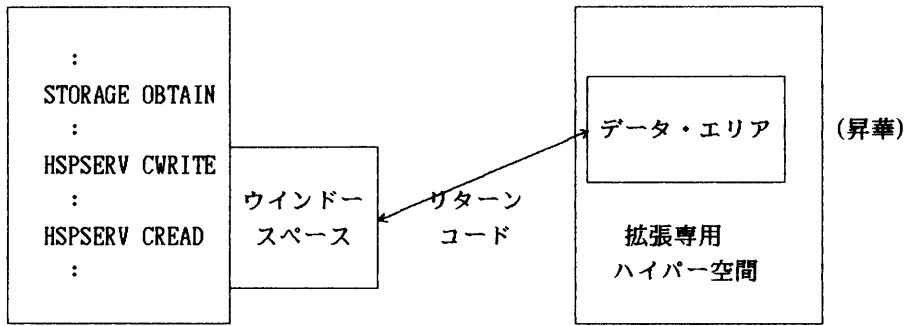
2.3 ハイパー空間

データ空間は中央記憶装置上の仮想空間であるがこのハイパー空間は拡張記憶装置上にとられる2ギガの空間でこの空間は直接命令によりアクセスできないため、バッファとしてウインドー・スペースを用意しハイパー空間とのデータの入出力(ページ移動)を実施しデータを格納、取り出しを行う。

このハイパー空間には下図のように拡張記憶装置のページ・スチール操作で外部ページ・データセットにページ・アウト(マイグレート)される標準ハイパー空間とページ・アウトされずに、昇華モードとして捨てられてしまう拡張専用ハイパー空間の二種類がある。



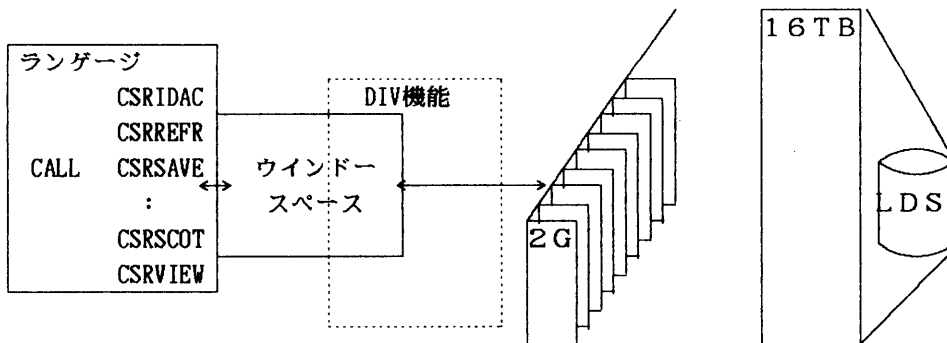
第三図: 標準ハイパー空間



第四図：拡張専用ハイパー空間

2.4 データ・ウィンドー サービス

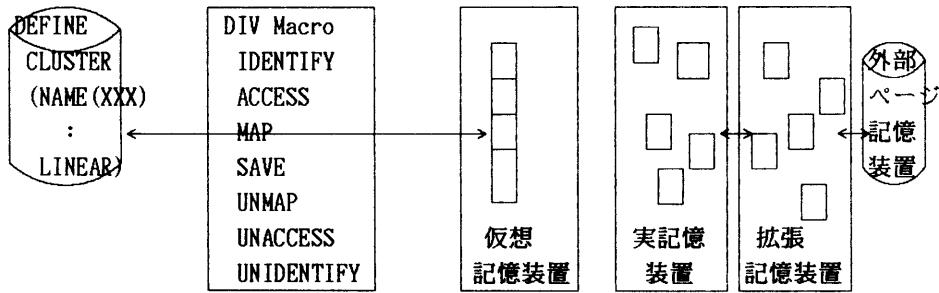
標準ハイパー空間をさらに使いやすくする機能がデータ・ウィンドー サービスでユーザはアセンブラだけでなくFORTRAN, COBOL, PL/I, PASCAL等の高級言語からCALLインターフェースで呼ばれ、一空間が最大16テラバイトまで使用でき、次に述べるデータ・イン パーチャル機能(DIV)もこのなかに組みこまれており巨大なファイルを簡単に操作できる機能である。



第五図：データ・ウィンドウ・サービス

2.5 データ・イン パーチャル機能

ユーティリティで形式化されたファイルを前以て用意しこのファイルをオープン後仮想記憶域上に確保されたウィンドー域と必要なレコード域をマップし、このファイルをページ・データセットとみなしてDIVマクロとページング機能をうまく使ってユーザ・プログラムに直接入出力要求無しに永久データセットが作れる機能であり、優しいプログラム開発、最大のシステムスループウトを狙ったファイル構造の中心となる機能である。この機能はMVS/XAでも使える機能であるがESAではデータ空間、ハイパー空間がこのウィンドー域として使え、データ常駐機能が本当の意味で可能となる。

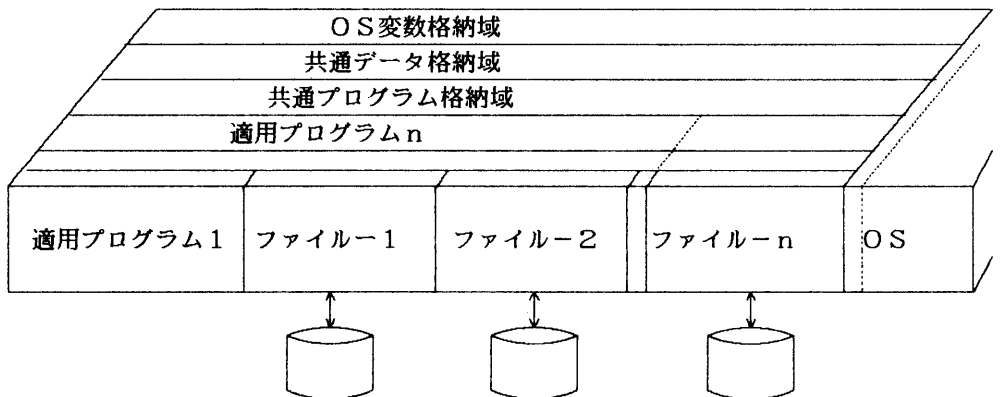


第六図：データ・イン・パーチャル機能

3. データ・イン メモリー

3. 1 データ・イン メモリーの一例

これまでの適用プログラムはファイルからデータを読みこみ本来の機能が活動するのが一般的であるが、殆どの場合このプログラムの開発上一番難しく、時間の要した部分がこの入出力の部分である。これを前節で述べたデータ・イン パーチャル機能をデータ空間に使うことによりファイルあたり2ギガのプログラム定数域が定義されたこととなる。即ちプログラムはデータがどこにあり、どの時点でデータを自分のプログラム内にとりこむかを考える必要がなく、常に定数、変数のためのデータ域がこのプログラム域の延長上に一体化されていると思えばいいわけである。

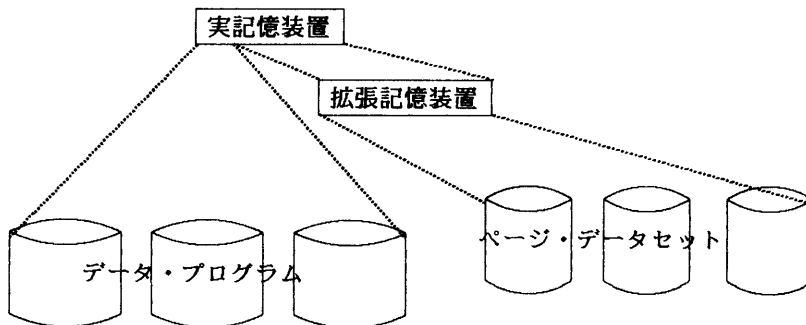


第七図：プログラム構造

3. 2 データ・イン メモリーの意義

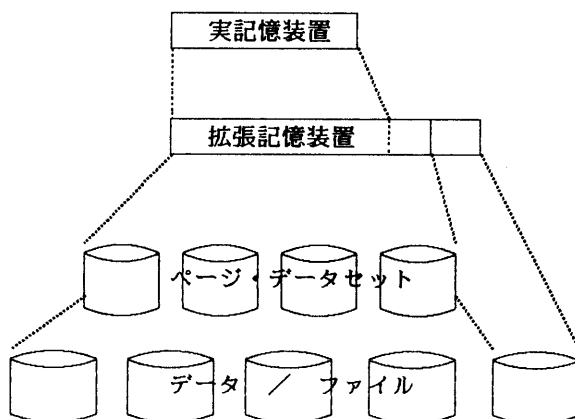
基本的にこれまでの適用プログラムはそのプログラムのデザインの範囲内で最良のプログラムロジックを作るため種々のアクセスメソッドやソート等の入出力操作やデータの扱いを容易にする道具を使っ

ていた。しかしこの方法には限界があるばかりではなく、システム全体として必ずしも最適な方法とは言いきれないばあが多く、システム資源の本当の意味での有効利用とならない場合が多かった。



第八図：これまでのOS上でのデータの動き

MVS/ESAでは必要なファイル全体をページング・メカニズムにのせ本当に必要なレコードだけを、使う頻度にしたがってよりうえの高価な場所に置いてくれる訳であり、システム資源全体がデータを保管アクセスするためのものとなるわけである。そして同じデータを再び読むことが大きく削減され、本当に必要な入出力だけになり、全システム資源を十分に活用出来ることとなる。



第九図：MVS/ESAによる資源の活用

おわりに

MVTや他OSと比較してMVSの巨大なスループットを出す能力を持っている根源はプログラムの参照の局所性を基礎とした仮想記憶の機能にある。MVS/ESAはデータの参照域の局所性を基礎とした仮想記憶機能、空間機能をスループット巨大化の基礎としておりこれに沿ったプログラムデザインをすることによりそのシステムがもっている本来のシステムスループットが得られる事となる。