

7G-5

バッチ処理の高速化 — エクセルバッチ —

村川賀彦, 杉山悦男, 大西俊也

㈱富士通静岡エンジニアリング

1. はじめに

現在、金融・流通・製造などさまざまな業種のホストコンピュータではオンライン処理だけでなく、オンライン処理の集計などのデータの一括処理としてバッチ処理が行われている。近年、業務の拡大やEDP化の進行により、バッチ処理で扱うデータ量が急増し、それにともないバッチ処理時間も延びてきている。また、オンライン時間の延長により、主としてオンライン処理を行っていない夜間に実行されるバッチ処理の時間を短縮することは急務となっている。これまでも、各種チューニングにより短縮を図っていたが、それも限界となり新方式での解決が必要となってきた。

そこで、バッチ処理の高速化を図る新方式としてエクセルバッチを開発した。本稿では、エクセルバッチの処理方式、その実現方式の概略と性能評価について述べる。

2. 処理方式

エクセルバッチは、バッチ処理がプログラム間で一時的なファイルを介したデータの受け渡しに時間がかかる事に着目して、データ処理の高速化と並列化を図ったものである。

従来のバッチ処理方式を図1に示す。この図より分かるよ

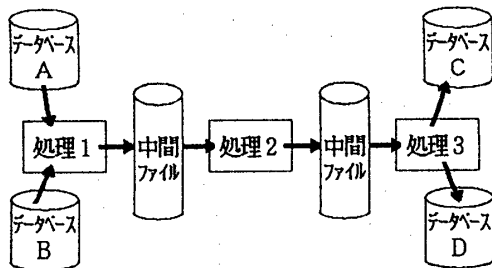


図1 従来のバッチ処理方式

うに、一つの業務を複数の処理に分割し、各処理間で磁気ディスク装置上の中間ファイルを経由しデータを引き継いでいた。この方式には、以下の2つの問題点があり、処理全体の実行時間が短縮できなかった。

- ①中間ファイルの作成側の処理が完了しないと、次の処理が開始できない。
- ②磁気ディスクを用いるため、入出力にかかる時間が大きい。

エクセルバッチではこの2つの問題点を以下の処理方式で解決した。

①並列実行

バッチ処理の多くのプログラムでは、まず先行処理で、ある処理単位(レコード)毎にデータを処理し、中間ファイルに順次書き込む。次に後続処理で、レコード単位にデータを中間ファイルより順次読み込み、処理を行っている。つまり、すべてのデータが中間ファイルに書き込まれていなくても、先行処理がレコードを書き込んだ時点で、後続処理は動作が可能となるわけである。

エクセルバッチでは、この点に着目して、各処理を独立した実行単位(ジョブ)として動作させ、それらのジョブ間で動的にデータを引き継ぐことによって、各処理を並列に動作させることとした。

②システム記憶(拡張かつ共用記憶)の利用

磁気ディスクで並列実行を可能とすると、かえって入出力の負荷が増大してしまい、経過時間の短縮にはならない。

エクセルバッチでは、システム記憶を介したデータ引き継ぎを行い、入出力時間を大幅に削減した。システム記憶は、CPU命令で主記憶との高速データ転送が可能であり、データ転送の待ち処理も不要であるため、入出力処理が非常に高速となる。また、システム記憶は、複数システム間で共用が可能であるため、エクセルバッチでも複数システム間でのデータ引き継ぎを可能としている。

エクセルバッチのバッチ処理方式を図2に示す。

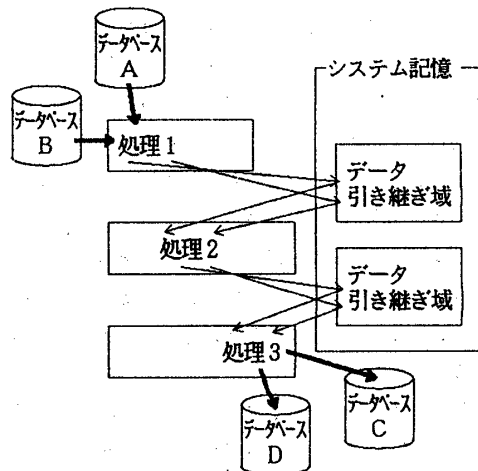


図2 エクセルバッチのバッチ処理方式

3. 実現方式

エクセルバッチを実現するための、①システム記憶を介したデータの引き継ぎとその同期制御、②並列実行にともなうジョブの制御の概要について述べる。

①データの引き継ぎとその同期制御

エクセルバッチにおけるジョブ間のデータ引き継ぎを図3に示す。ジョブ1が先行処理、ジョブ2が後続処理である。

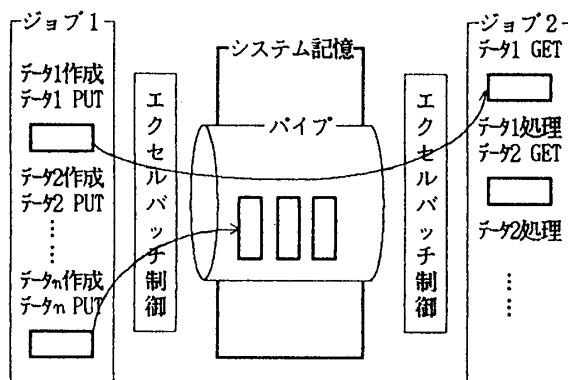


図3 データの引き継ぎ

ジョブ1が作成したデータ1は、エクセルバッチ制御により、システム記憶上のデータ引き継ぎ域（以下、パイプ）に書き込まれる。もし、それ以前にジョブ2がデータの読み込みを要求していれば、パイプ内にはデータがないので、データが書き込まれるまでエクセルバッチ制御により待ちとなっている。データ1が書き込まれると待ちが解除され、データ1がジョブ2に渡る。ジョブ2がデータの読み込み要求をしていなければ、データ1はパイプ上に一時的に保存される。また、ジョブ2がなかなか読み込みを開始しないと、パイプがデータでフルとなる事もある。この時、ジョブ1はエクセルバッチ制御により待ちとなり、ジョブ2が読み込みを開始すると待ちが解除される。

データの入出力には、既存のデータ管理の順次アクセス法がそのまま使用できる。よって、プログラムの変更は必要なく、エクセルバッチが利用できる。また、読み込まれたデータはパイプ内より削除されるため、全データをシステム記憶上に置くことに比べて、システム記憶を有効に利用できる。

②並列実行にともなうジョブの制御

エクセルバッチを使用するには、1つの業務を構成する各処理を、独立した実行単位のジョブに分割し、それを1つにまとめたジョブ群として入力し、各処理をジョブとして並列に動作させる。

それぞれのジョブのジョブ制御文（JCL）には、そのジョブが属するジョブ群と、ジョブ間のデータ引き継ぎにエクセルバッチを使用するファイルを定義する。エクセルバッチ制御は、各ジョブの定義をもとに、ジョブ間の引き継ぎ関係を把握し、ジョブ間でデータを引き継ぐためのパイプをシス

テム記憶上に確保する。これは、使用するジョブがすべて終了したら、消去される。

また、ジョブ群のあるジョブが異常終了すると、エクセルバッチ制御がそれを他のジョブに通知し、異常終了とする。これは、データの引き継ぎで同期処理をおこなっているため、書き込み側あるいは読み込み側のジョブが異常となってしまうと、他ジョブはいつまでも待ち状態となってしまうためである。

4. 性能評価

実際の業務を単純化したものとして、図4に示すような処理で経過時間を測定してみた。（ソートは多くの業務処理で一般的に使用されている。）

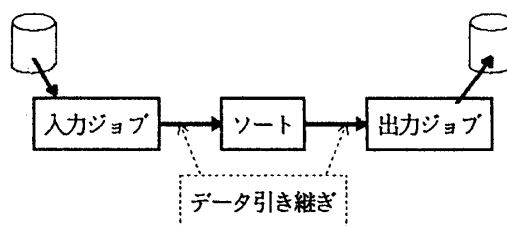


図4 処理例

データ引き継ぎを磁気ディスクで行った場合と、エクセルバッチを利用した場合のそれぞれの処理経過時間を測定して比較すると、図5のように約1/3となる場合もあった。

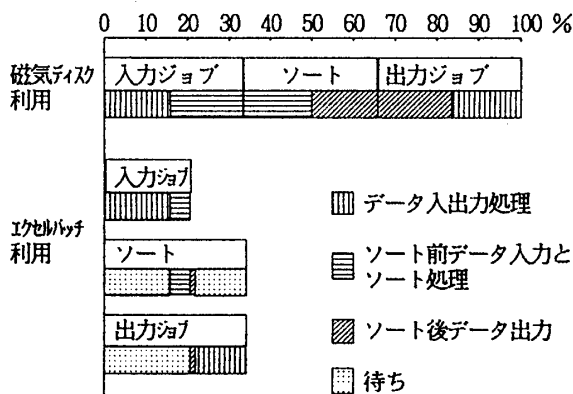


図5 経過時間の比較

また、実際の業務に適用した例としては、製造業の月次在庫管理業務に適用し、業務経過時間を2/3～1/2とすることができた。

5. おわりに

以上バッチ処理の高速化の新方式であるエクセルバッチについて述べた。今後は、さらに運用方法を改善し、使い易くしていきたいと考えている。

参考文献

[1] 中島 他, "FUJITSU M-1800Fグループのソフトウェア:MSP-EX," FUJITSU, 42, 2, pp. 139-150 (1991)