

実行多重度の向上による  
バッチ処理高速化機能の開発

6M-8

1 渡辺和彦 1 長須賀弘文 2 杉田潔

1 (株)日立製作所 2 日立ソフトウェアエンジニアリング (株)

1. はじめに

大規模計算機システムでは、オンラインの稼働時間延長に伴い、データを集計するバッチ処理を実行できる時間帯が制約を受ける。オンラインの処理量の増加やサービスの多様化に伴い、バッチ処理量が増大し処理形態も複雑化しつつあるため、バッチ処理全体を抜本的に見直し、高速化が求められている。そこで、一つのジョブの中の複数のジョブステップを並列実行させることで実行多重度を向上させて処理時間の短縮を図るバッチ処理高速化機能を開発した。本稿では機能の構成と効果を報告する。

2. バッチ処理の構成

一般にバッチ処理は数多くのジョブで構成する。さらに個々のジョブは複数のプログラムの実行単位（ジョブステップ）を複数組み合わせる構成する。ジョブは上段のジョブステップから一段ずつ逐次的に実行されている。その中には逐次的に実行する必要のないジョブステップやPREST<sup>[1]</sup>というパイプを使い、並列に実行できるジョブステップが含まれる。

3. バッチ処理高速化機能

3.1 ジョブステップ並列実行機能

ジョブ制御言語に並列実行の指定があると、指定のあったジョブステップを自動的に分割して別のジョブとして実行することによりジョブステップの並列実行を実現する。また、実行中の操作を一元化し、実行終了後に分離したジョブを一つのジョブにまとめることで一つのジョブとし

て運用できる機能である。（図1参照）

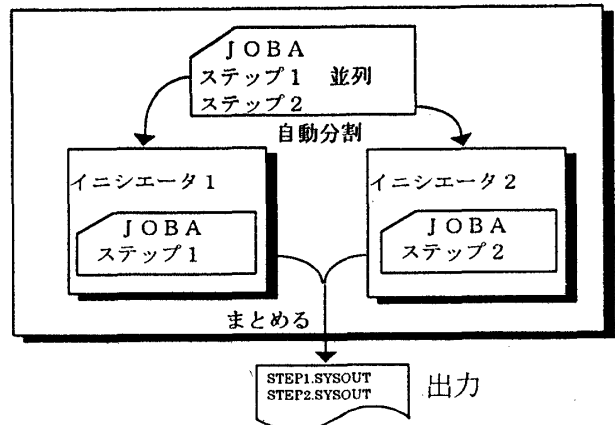


図1 ジョブステップ並列実行機能

3.2 適用支援機能

並列実行するにはジョブステップの関係を熟知した熟練者がいなければ並列指定をすることは困難である。そこで、並列実行できるジョブステップを以前に実行した稼働履歴から、ジョブステップ間の関係を判定して、熟練者でなくても並列指定が可能なジョブステップを抽出する方式を開発した。さらに、並列実行効果を量的に見積もれる機能も併せて開発した（図2参照）。

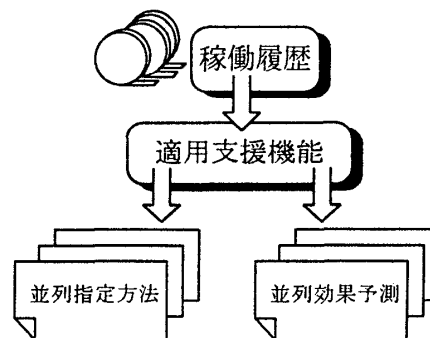


図2 適用支援機能

A development of Job-step Parallel Execution Control facility, depend on multiplex execution.

1Kazuhiko Watanabe, 1Hirofumi Nagasuka, 2Kiyoshi Sugita

1Hitachi, Ltd. 2Hitachi Software Engineering Co., Ltd.

## 4. 効果

### 4.1 バッチ処理高速化機能の効果

金融系システムを適用支援機能で解析した結果、前後の関連のないジョブステップが45%、PRESTにより並列化できるジョブステップが35%存在することを確認できた(図3参照)。機能適用例として、実行時間を24%に低減したバッチジョブの実行時間と適用支援機能の予測実行時間を表1に示す。

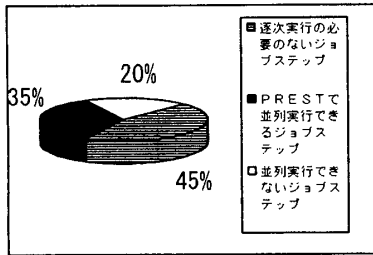


図3 ジョブステップ解析結果

表1 並列実行時間予測と実行時間

ステップ	逐次実行時間	バッチ処理高速化機能		
		並列	予測時間	実行時間
STEP1	0:00:01		0:00:02	0:00:01
STEP2	0:05:05	並列	0:05:05	0:05:06
STEP3	0:06:12	並列	0:06:12	0:06:13
STEP4	0:02:03	並列	0:02:04	0:02:02
STEP5	0:02:46	並列	0:02:46	0:02:46
STEP6	0:04:06	並列	0:04:07	0:04:07
STEP7	0:04:08	並列	0:04:08	0:04:07
STEP8	0:01:02	並列	0:01:02	0:01:02
STEP9	0:01:10	並列	0:01:10	0:06:10
STEP10	0:00:00		0:00:01	0:00:01
合計	0:26:38		0:06:15	0:06:17

時間単位: h(時):min(分):s(秒)  
 ジョブステップ間のプログラムの関連がない場合の測定値

### 4.2 多重度向上例

機能適用例を図4に示す。並列実行すると実行時間は短縮できる。しかし、個々のジョブステップが利用できる資源が減少するために応答時間が悪化することがある。そこで応答時間を保証するために、その時に実行しているジョブステップ数をインシエータ数以上に動かさないように制御する。ある計算機システムでは前後の関連のないジョブステップは全体の45%もある。このジョブステップの実行はインシエータ数に空きのある場合だけバッチ処理を高速化できるように多重度を向上して処理時間の短縮と応答性能を

確保できる。図4ではインシエータ数が3つの場合と4つの場合を示している。3つの場合は4つの場合より処理時間が増加するが応答時間は保証できるようになる。

## 5. まとめ

実行多重度の向上によりジョブの処理時間を短縮できる。ただし、無秩序に多重度を増加させてしまうとCPU資源を大幅に利用してしまい、他のジョブステップの応答時間に影響を与えてしまう。そこで、多重度の上限を管理しながら実行多重度を向上可能な方式を開発した。

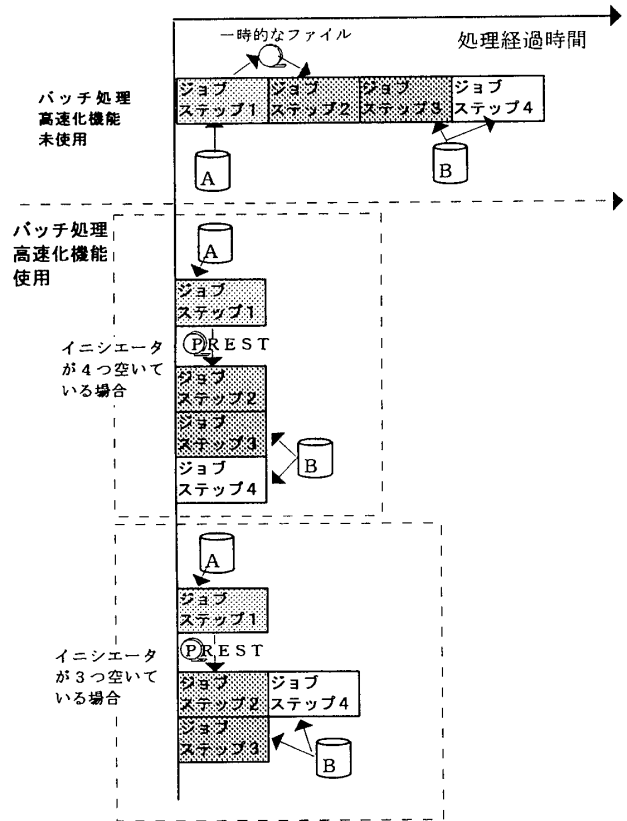


図4. バッチ処理高速化機能によるバッチ処理時間短縮

## 参考文献

- [1]長須賀, 他: ジョブ間並列同期転送機能の開発と評価-VOS3/AS:PREST-, 情報処理学会第41回全国大会講演論文集, 1990年9月。