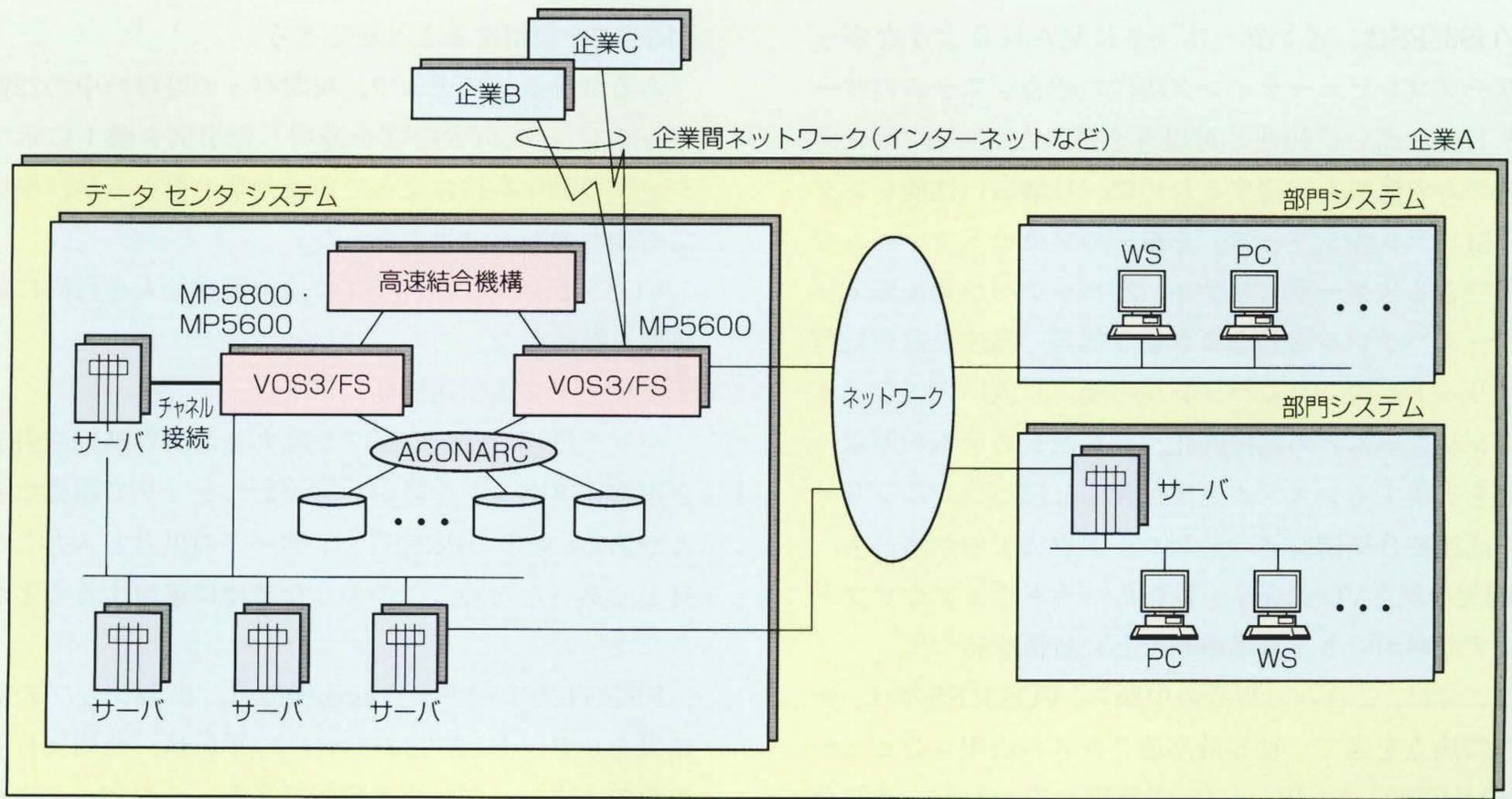


オペレーティングシステムによる基幹情報システムの構築支援—バッチ処理の時間短縮およびスループット向上—

Support of Creating a Mission Critical Information System with an Operating System

山岸 正* *Tadashi Yamagishi* 飯田恒雄* *Tsuneo Iida*
新村義章* *Yoshiaki Shimmura* 守島 浩* *Hiroshi Morishima*



注：略語説明
VOS3/FS (Virtual-storage Operating System 3/Forefront System Product)
ACONARC (Advanced Connection Architecture)

21世紀に向けた企業の基幹情報システムの構築

VOS3/FSは、圧倒的な処理能力でオンラインランザクション処理からデータベース処理やバッチ処理までこなせるデータセンタシステムの構築を支援するものである。

情報化や国際化の進展に伴い、インターネットに見られるようなネットワークコンピューティング時代の統合システムの重要性が増している。この統合システムの中の大規模業務を処理するサーバとして、圧倒的な処理能力でオンラインランザクション処理からデータベース処理やバッチ処理までこなせるデータセンタシステムが望まれている。このようなニーズにこたえて、多彩な並列処理でデータセンタシステムを実現する“VOS3/FS” (Virtual-storage Operating System 3/Forefront System

Product)を開発した。

VOS3/FSは、情報システムの構成が多様化しているため、(1)幅広い性能レンジの「Mパラレルシリーズ」を単一のプラットフォームで使用できる、(2)データベースパラレルやバッチパラレルなど多彩な並列処理機能の中から業務に最適なものを選択できる、(3)膨大なユーザー資産を継承できるなどの特徴を持つ。このVOS3/FSは、21世紀に向けた企業の基幹情報システムの構築に役立つものと期待できる。

* 日立製作所 ソフトウェア開発本部

1 はじめに

情報化や国際化の進展に伴い、国際金融決済や通信の料金計算、航空機の座席予約などに見られるような大規模業務の増大に対応して、圧倒的な処理能力でオンライントランザクション処理からデータベース処理やバッチ処理までこなせるデータセンタシステムの構築が求められている。

VOS3/FSは、インターネットに見られるようなネットワークコンピューティング時代の統合システムのサーバとして、高い信頼性と可用性を持ったデータセンタシステムの構築を支援するために、(1)幅広い性能レンジの「Mパラレルシリーズ」を単一のプラットフォームで使用できるスケラビリティ、(2)バッチパラレルやデータベースパラレルなどさまざまな業務で高速な並列処理のメリットが得られるパラレル機能、(3)PC・WSなどオープンシステムとの連携強化、(4)複数システムの運用一元化を実現するシステム運用強化、(5)Mパラレルシリーズや高速結合機構などの最新ハードウェアのサポート、(6)膨大な既存VOS3資産を最新のハードウェアやソフトウェアで利用できる資産継承などの特徴を持つ¹⁾。

ここでは、これらの特徴の中から、VOS3/FSのバッチ処理に焦点を当て、世界最高速クラスの汎用コンピュータ“MP5800”のCPUパワーを利用して、(1)バッチ処理時間の短縮をねらえるバッチパラレル機能、(2)高速結合機構を利用したMパラレルシリーズを使用してバッチ処理のスループット向上がねらえるシステム間パラレル機能、(3)オンライントランザクション処理やデータベース処理を行うサブシステムが、高速結合機構を利用したパラレル処理を実現するために提供するパラレル基本機能について述べる。

2 バッチパラレル機能

2.1 並列同期転送機能

バッチ処理では、ジョブ間やジョブステップ間の実行結果の引き継ぎを、磁気ディスクや磁気テープ装置上の順編成データセットを使用して行うことが多い。この引き継ぎ用のデータセットを仮想記憶装置上に配置して、実行結果の引き継ぎをデータ転送で実現する機能がPREST(Parallel Reference and Synchronous Transfer Facility：並列同期転送機能)である。

磁気ディスクや磁気テープ装置への入出力が不要になることと、データセットへ出力するジョブとデータセッ

トから入力するジョブの並列実行が可能になることにより、バッチ処理時間を短縮することができる。データ転送は、あらかじめ設定したメモリ容量内で入出力を同期させながら行うため、従来発生していたデータ量の増加によるデータセットの容量不足が回避でき、ジョブの中断、再実行が不要になる。引き継ぎ用データセットは、従来と同様にデータセットとして見えるので、プログラムを修正することなくジョブ制御文の指定だけでPRESTを使用することができる。

ある金融系システムで、夜間バッチ処理の中の22個のバッチジョブにPRESTを適用した事例を表1に示す。ジョブの動作特性によって短縮効果が異なるが、平均して約50%の短縮効果があった。

VOS3/FSでは、PRESTの適用業務拡大を目的に次の機能を拡張した。

(1) データ多重転送機能

バッチ処理では、ジョブが出力した実行結果を引き継ぎ用データセットを経由して複数のジョブで編集することがある。従来のPRESTは、データの出力と入力とが1対1であったため、このような業務に適用することができなかった。

PRESTのデータ多重転送機能は、出力ジョブの実行結果をレコード(またはブロック)単位に、並列実行中の複数の入力ジョブへ渡す機能である。このデータ多重転送機能を図1に示す。出力ジョブのジョブ制御文に、データ多重転送機能の指定と並列に入力するジョブ数を指定することによってこの機能が使用できる。

データ多重転送機能では、出力ジョブが出力したレコードを、あらかじめ指定したすべての入力ジョブが入力したことを確認するように制御する。そのため、それぞれの入力ジョブは、出力ジョブが出力したすべてのレコードを確実に受け取ることができる。

表1 PRESTの適用事例

ある金融系システムで、22個のバッチジョブにPRESTを適用した結果、約50%のバッチ処理時間の短縮効果があった。

業 務	ジョブ数	適用前	適用後	短縮時間	効 果
当座業務 マージ処理	6 ジョブ	64分	35分	29分短縮	-45%
ジャーナル マージ処理	4 ジョブ	58分	10分	48分短縮	-83%
会計処理	6 ジョブ	52分	35分	17分短縮	-33%
バッチデータ 作成処理	6 ジョブ	142分	81分	61分短縮	-43%

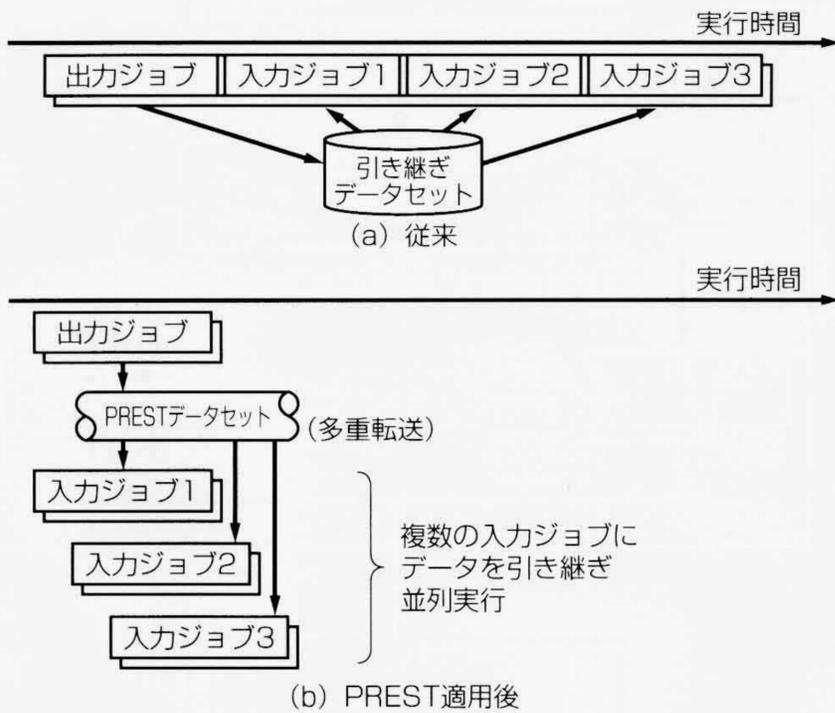


図1 PRESTのデータ多重転送機能
出力ジョブの実行結果をレコード(またはブロック)単位に、並列実行中の複数の入力ジョブへ渡すことにより、ジョブの実行時間が短縮できる。

(2) データ分割転送機能

データ分割転送機能は、出力ジョブの実行結果をレコード(またはブロック)単位に、並列実行中の複数の入力ジョブに分割して渡す機能である。入力ジョブから入力要求を受けた順番にレコードを渡していくことにより、入力ジョブ間の待合せを排除し、分割処理の高速化を実現している。データ多重転送機能と異なるのは、引き継ぎデータセット内の各レコードを、いずれか一つの入力ジョブにだけ渡す点である。出力ジョブのジョブ制御文で、データ分割転送機能の指定と並列に入力するジョブ数を指定することによってこの機能が使用できる。

この機能を大量のソートマージを行うジョブに利用すると、ソートマージの対象レコードを幾つかに分割し、分割したレコードごとに並行してソートを行い、その結果をマージすることができるのでバッチ処理時間が短縮できる。

(3) データセット同時出力機能

データセット同時出力機能は、出力ジョブの実行結果をPRESTを使用して入力ジョブへ渡すと同時に、外部記憶装置上のデータセットへも出力する機能である。この機能により、出力ジョブの実行結果を保存するためのジョブを新たに作成するなどの作業が不要になる。出力ジョブのジョブ制御文に、データセット同時出力機能の指定とデータを出力する外部記憶装置上のデータセットを指定することによってこの機能が使用できる。

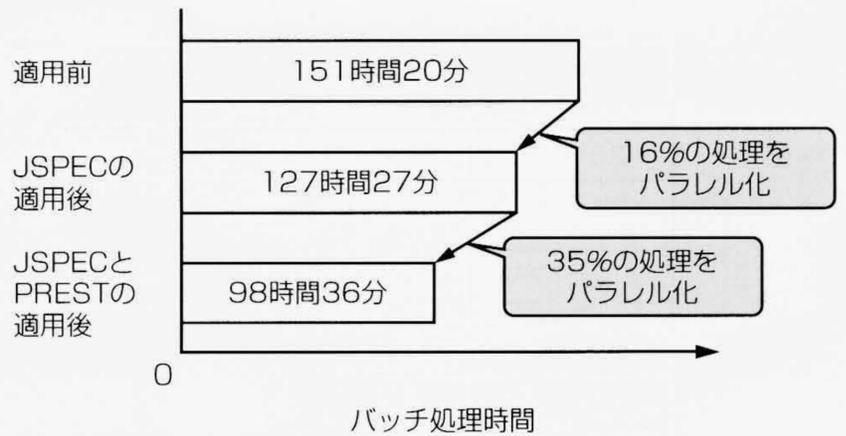


図2 JSPECの適用効果の見積もり
ある金融系システムで効果予測をした結果、JSPECの適用で16%、さらにPRESTの適用で35%の短縮が見込めることがわかった。

2.2 ジョブステップ並列実行機能

JSPEC(Job Step Parallel Execution Control Facility: ジョブステップ並列実行機能)は、ジョブの中で独立しているジョブステップや、PRESTが適用できるジョブステップを並列実行することにより、バッチ処理時間の短縮を実現する機能である。

ある金融系システムのバッチ処理にJSPECを適用した場合の効果予測を図2に示す。独立なジョブステップにJSPECを適用して各ジョブステップを並列実行することにより、1日のバッチジョブの実行時間の合計が16%短縮でき、さらにPRESTを適用することで35%の短縮が見込める。

2.2.1 JSPECのねらい

従来、一つのジョブを複数のジョブステップで構成している場合、それぞれのジョブステップを先頭から順番に実行していた(図3のステップ1からステップ3)。ところが、ジョブステップによっては必ずしも順番に実行する必要がないものもある。例えば、幾つかの先行ジョブステップでそれぞれデータを生成し、最後のジョブステップ(ステップ3)でそれらのデータを集めて加工するような場合、最終ジョブステップだけを最後に実行すればよく、先行ジョブステップ(ステップ1とステップ2)はそれぞれ並列に実行してもよいはずである。さらに、先行ジョブステップで出力したデータを後続ジョブステップへ受け渡している処理にPRESTを適用することにより、それらのジョブステップを並列に実行できる場合もある(ステップ1, ステップ2, ステップ3)。

2.2.2 ジョブステップの並列実行方式

並列に実行させたいジョブステップのジョブ制御文にパラレル指定を追加すると、JSPECはジョブ実行時にそのジョブステップを自動的に分割し、別々の仮想空間で

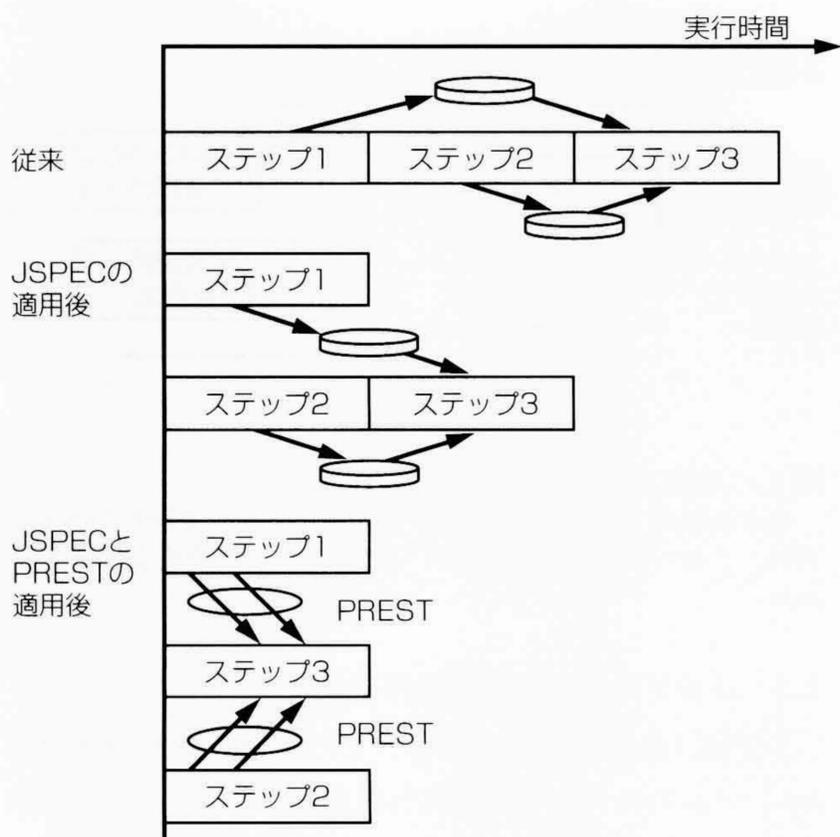


図3 JSPECのねらい

ステップ1とステップ2が独立している場合は、そのステップを並列実行させる。さらに、PRESTが適用できる場合は、ステップ3も並列実行させて実行時間を短縮する。

並列実行させる(図4参照)。

VOS3/FSは多重仮想記憶方式のため、それぞれのジョブステップごとに広いアドレス空間が使用でき、ジョブステップを並列実行することによってジョブの実行時間を短縮することができる。また、(1)ジョブのキャンセルやジョブの状態表示などを1回のコマンド投入で行えること、(2)各ジョブステップの出力結果を1ジョブにまとめることなどにより、単一ジョブイメージ操作を可能にした。

2.2.3 PRESTの適用拡大

JSPECを使ってジョブステップ間のデータ受け渡しにPRESTを適用することにより、従来のようにジョブに分割してジョブ間でPRESTを適用することに比べて、(1)簡単な指定だけでPRESTが適用でき、JCL(Job Control Language)を作り直す必要がない、(2)従来どおり1ジョブとして扱えるので運用管理を変更する必要がない、(3)並列ジョブステップ間の同期をとりやすいなどの効果がある。

2.3 バッチパラレル適用支援機能

バッチパラレルの適用時には、(1)適用可能ジョブの抽出、(2)JCLの書き換え、(3)適用後のジョブの検証が必要である。適用可能ジョブの抽出を支援するために、現行の稼動環境のシステム稼動履歴情報から、ジョブステッ

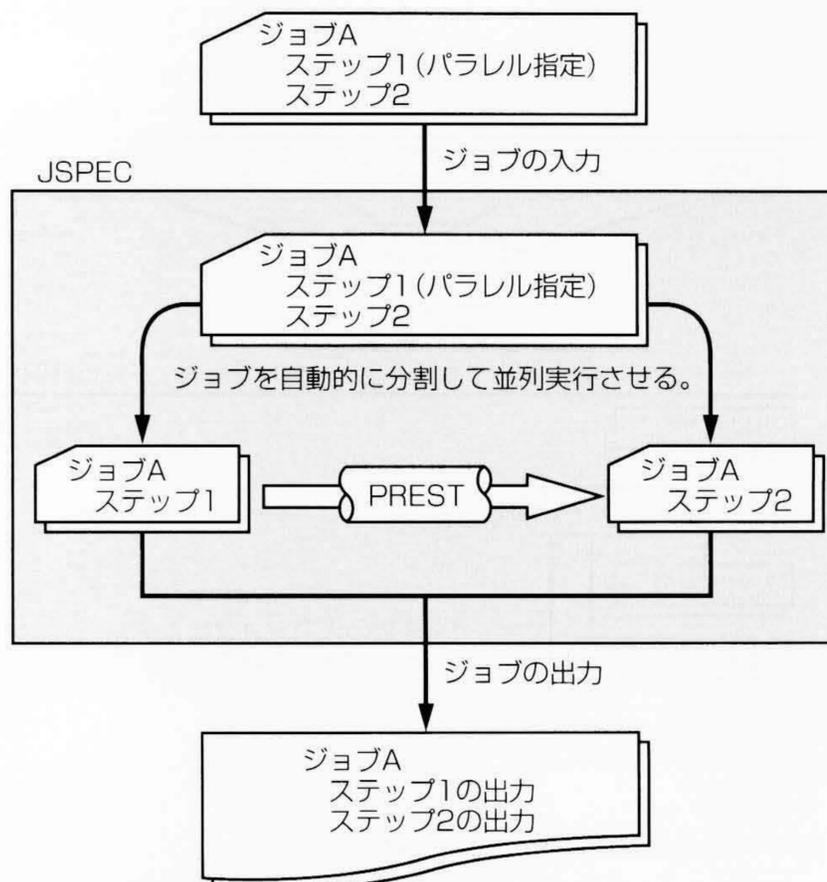


図4 ジョブステップ並列実行機能

ジョブ制御文にパラレル指定があると、そのジョブステップを自動的に分割して並列実行する。

ブ間の関係やデータセットのアクセス関係を判定して、JSPECおよびPRESTの適用可否、適用した場合のジョブの実行時間の短縮効果の見積もりが可能なバッチパラレル適用支援機能も合わせて開発した。図2に示すJSPECの適用効果の見積もりは、この機能を使用して短縮効果を予測した結果である。

3 システム間パラレル機能

従来の密結合マルチプロセッサでは、主記憶共用による排他オーバーヘッドのために、命令プロセッサ数の増加に比例した性能を確保していくことが困難であった。そのため、将来のデータ処理量の増加に対して情報システムのパワーに限界が生じる可能性が考えられる。

システム間パラレル機能は、システム間の負荷情報の管理、制御情報の高速通信、共用資源の排他制御などを専用のハードウェアである高速結合機構を使用を行い、多量のデータ処理を複数のシステムで分担して行えるようにする機能である。

3.1 システム間負荷分散ジョブスケジューリング機能

3.1.1 複数システムへのジョブの分散

パラレルシステムでは、特定のシステムに負荷が集中して、システムトータルとしてのパフォーマンスを低下させることがないように、負荷を分散する必要がある。

システム間負荷分散ジョブスケジューリング機能は、ジョブを各システムへ分散するときに、高速結合機構を利用して各システムの負荷(メモリ使用状況やCPU使用率)を考慮しながら最適なシステムへ自動的にジョブを分散する機能である(図5参照)。この機能を利用すると、システム間の負荷の偏りをなくし、パラレルシステム全体としてのスループットの向上が実現できる。また、ジョブが使用する資源(メモリや装置)を考慮しながらジョブをスケジューリングする総枠制御ジョブスケジューリング機能により、資源をむだなく使用することができる。

3.1.2 任意のシステムでのジョブの実行

ジョブが使用する磁気テープ装置の使用順序をシステム間で制御する磁気テープ装置共用機能、データセットのシステム間排他制御を行うデータセット共用機能、各システムのスプールを共用可能にするスプール共用機能により、任意のシステムでのジョブの実行を可能にした。また、各システムで実行したジョブの出力結果を任意のシステムで取り出すことができる。

3.1.3 システムの連続運転

24時間の連続運転に対応して、システムの起動・停止が発生しても、現在動作中のシステムの中で最適なシステムを選択してジョブを実行するので、動作中のシステムを意識することなくジョブの実行依頼ができる。

3.2 システム間バッチパラレル

従来、複数のVOS3/FSシステム間でのデータの引き

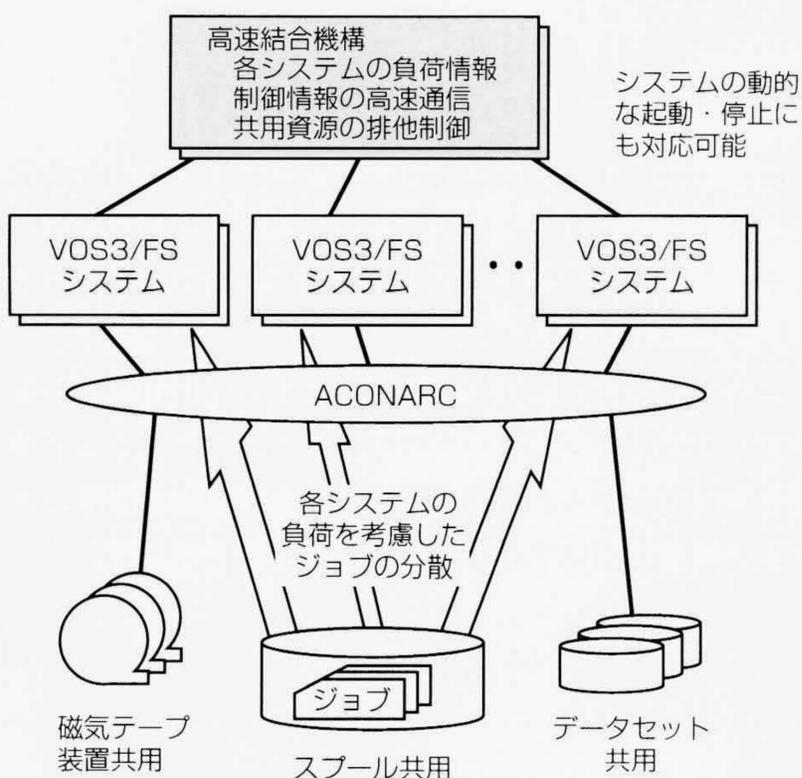


図5 システム間負荷分散ジョブスケジューリング機能
高速結合機構を利用して、各システムの負荷を考慮しながら最適なシステムへ自動的にジョブを分散する。

継ぎは、共用した外部記憶装置上に引き継ぎ用のデータセットを配置して実現していた。

この処理を高速に行うために開発したシステム間バッチパラレル機能を図6に示す。システム間PRESTは、PREST用の引き継ぎ領域を高速結合機構上に配置することにより、システム間の並列同期転送を実現した。システム間JSPECは、スプール共用機能を利用して他システムにジョブステップを転送することにより、システム間のジョブステップの並列実行を実現した。

この機能により、例えば勘定系システムと情報系システム間など、異なる業務システム間でデータを高速に引き継ぐことができる。また、複数システムにジョブステップが分散できるようになるため、1システムのCPU能力を越えてジョブステップの並列度を上げることができ、ジョブの実行時間を短縮することができる。

4 パラレル基本機能

VOS3/FSのシステム間パラレル機能をはじめ、オンライントランザクション処理やデータベース処理を行う各サブシステムが高速結合機構を利用したパラレル処理を実現するために、図7に示すパラレル基本機能を提供している。

4.1 システム間キャッシュ共用機能

システム間の共通情報を各システムの仮想アドレス空間上にキャッシュ領域^{*)}として配置し、一つのシステムがあたかもキャッシュ領域を占有しているかのようにア

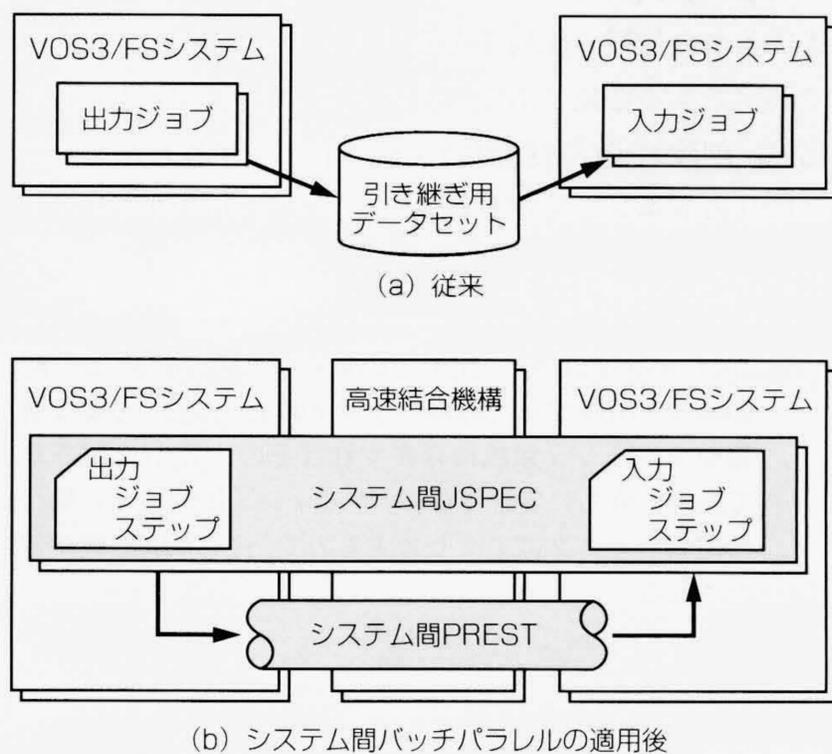


図6 システム間バッチパラレル機能
システム間PRESTとシステム間JSPECにより、複数システムを使用したバッチパラレルが可能になる。

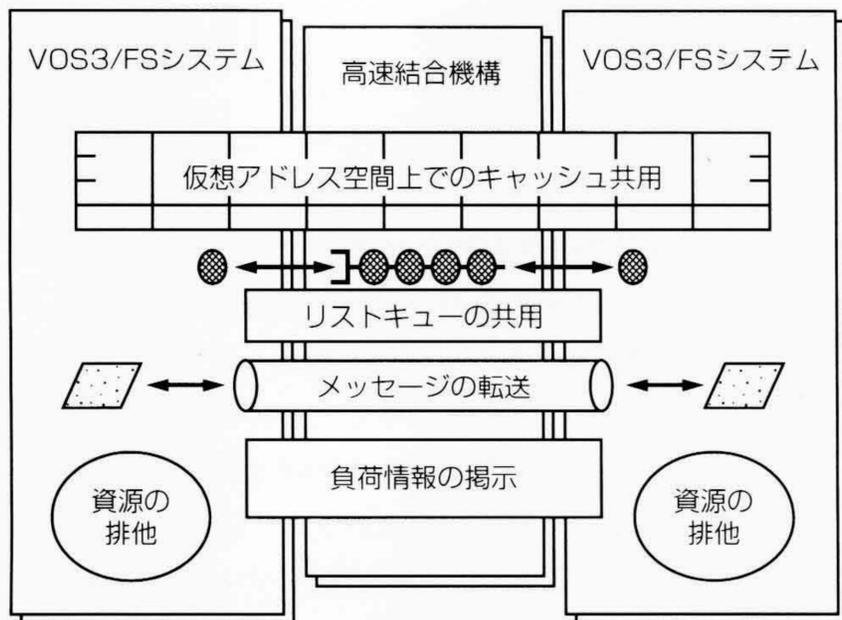


図7 パラレル基本機能

各サブシステムが高速結合機構を利用したパラレル処理を実現するための機能を提供している。

アクセスすることができる機能である。キャッシュ領域を一つのシステムが参照、更新する際に、キャッシュ領域ごとの専用ロックを確保することにより、他システムとの排他制御が可能になる。キャッシュ領域は通常の仮想アドレス空間上の記憶領域と同様に、任意の命令セットで参照、更新することができる。

この機能を利用すると、キャッシュ領域が磁気ディスク上のデータセットにアクセスするときのデータキャッシュとして利用できるため、システム間の大規模データベースの共用が可能になる。

4.2 システム間リスト共用機能

高速結合機構上に、複数のリストエントリをキュー形式に並べた情報の配列であるリストキューをシステム間で共用できる機能である。リストエントリの登録、取り出し、削除および移動ができる。また、リストキューごとにリストエントリが登録されたことを非同期にシステ

※) キャッシュ領域：主記憶と磁気ディスクのアクセス時間の差を吸収するためのバッファ領域のこと。必要な情報がキャッシュ領域に存在すればその内容が使用されて処理が直ちに実行される。キャッシュ領域にない場合は磁気ディスクにアクセスするので、その間、処理が待たされる。

ムから通知してもらうことができる。

この機能を利用すると、トランザクションキューのような各種リクエストキューを共用し、低負荷のVOS3システムがリクエストを取り出して処理するといった負荷分散を実現することができる。

4.3 システム間メッセージ通知機能

指定されたシステムにメッセージ(データ)を転送することができる機能である。送信先システムが受領する前に高速結合機構上にメッセージを登録すれば送信処理が完了するため、システム間で同期する必要がなく、システムスループットの向上が実現できる。この機能は、さまざまな付加機能を持った高度な通信機能ではなく、システム間の処理の引き継ぎや状態の通知といった、主にシステム内部で利用するものである。

4.4 システム間負荷分散機能

システムが連携して複数のリクエストを処理する場合、全システムのCPU利用率やメモリ使用量のような負荷情報を掲示し、システム間の負荷バランスを平準化することを支援する機能である。負荷情報には各サブシステムごとの負荷情報も掲示することができるため、サブシステムごとのきめ細かな制御に利用することもできる。

4.5 システム間ロック機能

システム間で取り決めた資源名を使った排他制御を高速に実現できる機能である。論理的な単位(例えばデータセット単位、もしくはブロック単位)での排他を実現しているため、排他のぶつかりを削減して並列動作率を向上することができる。

5 おわりに

ここでは、ネットワークコンピューティング時代の統合システムのサーバとして、高い信頼性と可用性を持ったデータセンタシステムを実現するVOS3/FSのパラレル機能について述べた。

このVOS3/FSは、既存VOS3システムで利用しているアプリケーション、データベースなどのソフトウェア資産を利用しながら、21世紀に向けた企業の基幹情報システムの構築に適用することができるものと考えている。

参考文献

- 1) 尾山, 外: 新世代システムによるデータセンタを実現するパラレルソフトウェア, 日立評論, 77, 5, 375~380(平7-5)