プロセス変身機能における資源の再利用効果

野村 和孝†　　田端 利宏††　　谷口 秀夫†††

†岡山大学大学院自然科学研究科
††九州大学大学院システム情報科学研究院
†††岡山大学工学部

分散環境において，実行途中のプロセスの実行環境を変更することができれば，負荷分散を実行することができ，有益である．このため，我々は，分散指向永続オペレーティングシステム ***Tender***（The ENduring operating system for Distributed EnviRonment）においてプロセスの実行環境を変更する機能であるプロセス変身機能を実現した．プロセス変身機能は，実行プログラムの変更，開始位置の変更，および動作空間の変更という三つの機能からなる．また，我々は，***Tender*** においてプロセスを構成する資源を再利用することにより，プロセスの生成処理と削除処理の高速化手法を実現した．本論文では，プロセスの構成資源を再利用したプロセス変身機能の高速化について，評価結果を報告する．

The effect by recycling of the resources in a process transformation mechanism

Kazutaka NOMURA † Toshihiro TABATA †† Hideo TANIGUCHI†††

†The Graduate School of Natural Sience and Technology,Okayama University
††Faculty of Infomation Sience and Electrical Engineering,Kyushu University
†††Faculty of Enginering,Okayama University

A process transformation technique has been implemented in ***Tender***. The purpose of this technique is load distribution in distributed environment. This technique consists of three functions, which are the change of a program, the change of start point of a program and the change of a virtual address space in process execution. In order to utilize this technique efficiently, it is necessary to reduce the cost of the transformation. In this paper, we discuss how a resource recycle mechanism is applied to the process transformation. We also evaluate the process transformation and report the result of the evaluation.

1　はじめに

複数の計算機を結んだ分散環境において，実行途中のプロセスを動的に他の計算機に移動できる機能を実現すれば，システム内で負荷分散を実現することができ，有益である．そこで，我々は，実行途中のプロセスの実行環境を変更する機能であるプロセス変身機能を，我々が開発している分散指向オペレーティングシステム ***Tender***（The ENduring operating system for Distributed EnviRonment）上に実現した [1].

また，プロセスの生成処理は，メモリ空間の生成やプログラムの読み込みなどの処理を必要とするため，OS の処理の中でも，負荷が大きい．このため，プロセスの生成処理と削除処理が頻発する応用プログラムでは，サービス処理性能が低下することがある．そこで，我々は，***Tender*** における資源の独立化機構を生かして，プロセスを構成する資源（以降，プロセス構成資源と呼ぶ）を再利用することにより，プロセスの生成と削除の処理を高速化することを実

現した [2].

本論文では，プロセス変身機能の高速化として，資源の再利用を行なう．プロセス変身機能である実行プログラムの変更，開始位置の変更，動作空間の変更におけるプロセス構成資源の再利用の実現方法を述べ，その効果について報告する．

2 *Tender* オペレーティングシステム

2.1 資源の分離と独立

Tender では，OS が操作する対象を資源として，分離し，独立化している．資源には，資源識別子と資源名を付与し，資源操作のインタフェースを統一している．各資源を操作するプログラム部品を資源ごとに分離し，共有プログラムを排除している．また，各資源の管理情報も資源ごとに分離し，各資源の管理表の間のポインタを禁止している．

このように資源の分離と独立化を行なうことで，プロセス構成資源を個別に変更することが容易になり，プロセスの実行環境を変更できる．プロセス構成資源は，各々独立しているため，プロセス構成資源を変更する際には，変更対象となる資源のみを扱うことができるので，変更処理を効率化できる．

また，資源の分離と独立化を行なうことで，資源の事前生成や保留が可能になり，資源の生成や削除を伴う処理を高速化することができる．

2.2 プロセス構成資源

プロセス構成資源を図 1 に示す．ここで，矢印は，資源の依存関係を表している．資源「プロセス」とは，プロセス識別子とプロセス管理表からなり，OS がプログラムの動作を制御する単位になる．したがって，プロセス生成時には，資源「プロセス」が，生成先計算機上に生成され動作を制御する単位になる．資源「プログラム」とは，実行プログラムのテキスト部とデータ部の大きさと先頭アドレス，およびプログラムの開始アドレスの情報からなり，プログラムの実行形式を隠蔽している．プログラムの内容は，資源「プレート」上に存在している．ここで，資源「プレート」とは，永続的な記憶を提供するものであり，既存の OS のファイルに相当する．

ここで，プロセス構成資源間の関係を図 2 に示す．資源「仮想空間」とは，仮想アドレスの空間であり，仮想アドレスを実アドレスに変換する変換表に相当する．資源「仮想領域」とは，メモリイメージを仮想化した領域であり，仮想領域の実体は，実メモリまたは外部記憶装置上に存在する．資源「仮想カーネル空間」，ならびに資源「仮想ユーザ空間」とは，プロセッサが仮想アドレスによって，前者はカーネルモードのみ，後者はユーザモードでもアクセス可能な空間である．両者は共に，仮想空間が持つアドレス変換表に，当該の仮想領域のデータ格納情報を設定する（以後，貼り付けると呼ぶ）ことで生成することができる．逆に，貼り付けた際に設定したデータ格納情報を解放する（以後，剥すと呼ぶ）ことにより，削除することができる．プロセスのテキスト部，初期値を持つ変数や文字列の集合部分であるデータ部，初期値を持たない変数や文字列の集合部分である BSS 部，およびユーザスタック部は，仮想ユーザ空間に読み込まれた状態で仮想空間上に存在する．

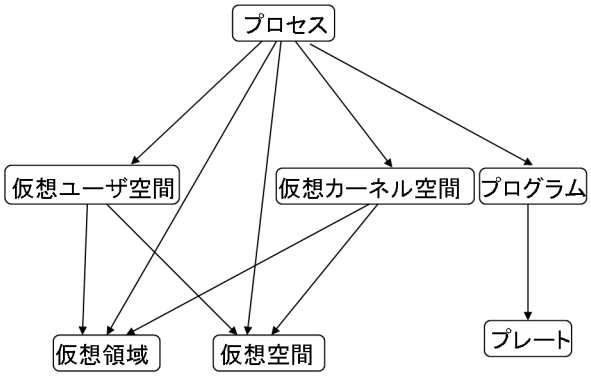

図 1 プロセス構成資源

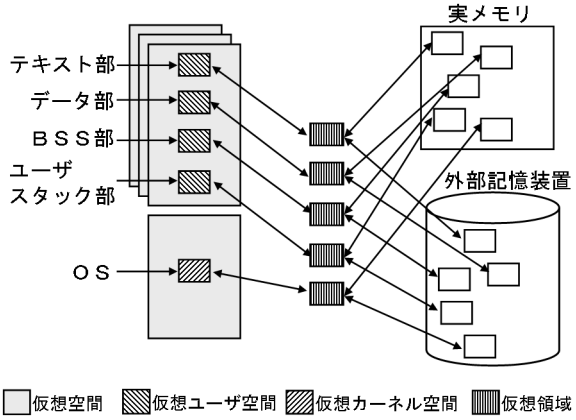

図 2 プロセス構成資源の関係

2.3 資源の再利用

Tender において，プロセスの生成処理では，OS

が再利用可能な資源を検索し，再利用可能な資源がある場合，プロセス構成資源を生成する処理を再利用処理に置き換えている．このように，プロセスの生成処理において，再利用可能なプロセス構成資源を再利用して，プロセス構成資源を生成する処理を短縮することにより，プロセス生成処理の高速化を実現している．

また，プロセスの削除時に再利用可能なプロセス構成資源を，再利用可能な資源を管理する管理表（以後，再利用可能資源管理表と呼ぶ）に登録し，実際に削除しないことにより，プロセス構成資源の削除に要する処理時間を短縮し，プロセスの削除処理の高速化を実現している．ここで，再利用可能資源管理表に登録された資源は，以後のプロセスの生成時にプロセス構成資源として再利用されることになる．

ここで，プロセス変身機能に関係のある再利用可能なプロセス構成資源について述べる．プログラムは，プログラムの内容が同じ場合，再利用できる．仮想空間は，一定の大きさの仮想アドレス空間を提供するアドレス変換表である．したがって，仮想ユーザ空間が存在しない仮想空間は，つねに再利用できる．仮想領域は，テキスト部，データ部，BSS 部，およびユーザスタック部について再利用可能であり，特定の大きさを持つ領域であるため，大きさの情報のみで再利用可能可否の判断できる．また，テキスト部についてはプログラムの内容が同じ場合にはテキスト部の仮想領域を共有することにより，テキスト部の内容も利用することができる．

2.4 プロセス変身機能

プロセス変身機能とは *Tender* においてプロセス構成資源を変更し，プロセスの実行環境を変更する機能のことである．プロセス変身機能として，以下の三つの機能がある．

（機能 1）実行プログラムの変更
（機能 2）開始位置の変更
（機能 3）動作空間の変更

以降に，これら三つの機能について述べる．

2.4.1 実行プログラムの変更

実行プログラムの変更とは，プロセスとして実行されるプログラムを変更する機能である．実行プログラムを変更したプロセスは，変更後のプログラムの先頭から処理を行なうこととなり，複数のプログラムを一つのプロセスとして逐次的に実行できる．

2.4.2 開始位置の変更

開始位置の変更とは，変更対象プロセスが次に走行する開始位置を変更する機能である．開始位置の変更には，プロセスを初期状態に変更する機能と，指定された位置に変更する機能がある．初期状態に変更する機能により，プロセスの再起動が可能になり，プロセスの生成と消滅のオーバヘッドを削除できる．指定された位置に変更する機能により，プロセスを保存した状態から再開させることができる．これにより，プロセスの実行途中で起こる障害からの高速な回復処理を実現できる．

2.4.3 動作空間の変更

動作空間の変更とは，プロセスが利用する仮想記憶空間を別の空間に変更する機能である．動作空間の変更には，動作空間を同一計算機内の別空間に変更する機能と，他計算機上の空間に変更する機能がある．同一計算機内の別空間に変更する機能により，プロセス間で協調処理を行なう際，処理が緊密な場合には，一方のプロセスの動作空間を他方のプロセスの空間へ変更して処理を行なうことで，空間切り替えによるオーバヘッドの削除が可能になる．他計算機上の空間に変更する機能により，負荷分散，協調処理のオーバヘッドの削除，および保守や点検による計算機の停止に伴うプロセス停止の回避が可能になる．

3 プロセス変身機能における資源の再利用

ここでは，プロセス変身機能である実行プログラムの変更，開始位置の変更，および動作空間の変更における資源の再利用方法について述べる．プロセス変身機能において再利用可能な資源は表 1 に示すとおりであり，以降に詳細を述べる．

3.1 実行プログラムの変更における資源の再利用

実行プログラムの変更の様子を図 3 に示す．まず，変更前に利用していた各部の仮想領域を剥し，変更後に利用する各部の仮想領域の貼り付けを行った後に，変更後のプログラムを読み込む．

従来の手法では，図 4 に示すように，変更後に利用するプログラムと，テキスト部，データ部，および BSS 部の仮想領域を新規に生成していた．そこで，これらの資源を再利用して利用することができるように，実行プログラムの変更を図 5 に示すように変更した．プログラムは，再利用可能なプログラムを検索し，再利用可能なプログラムが存在すればその

表 1 プロセス変身機能における再利用可能な資源

機能		再利用可能な資源	再利用可能資源管理表に登録可能な資源
実行プログラムの変更		プログラム，テキスト部の仮想領域データ部の仮想領域，BSS 部の仮想領域	テキスト部の仮想領域，データ部の仮想領域，BSS 部の仮想領域
開始位置の変更	初期状態	なし	なし
	指定された位置	なし	なし
動作空間の変更	同一計算機内	仮想空間	なし
	他計算機上	なし	なし

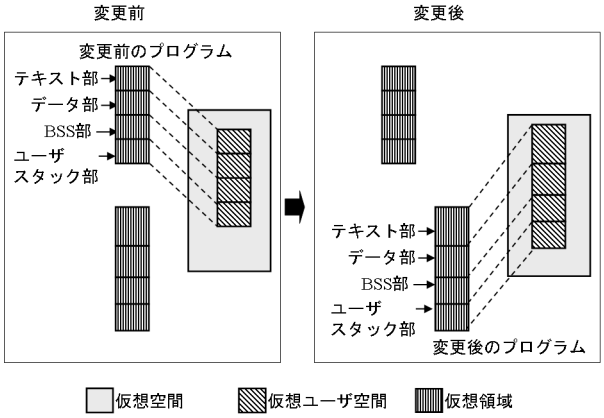

図 3 実行プログラムの変更の様子

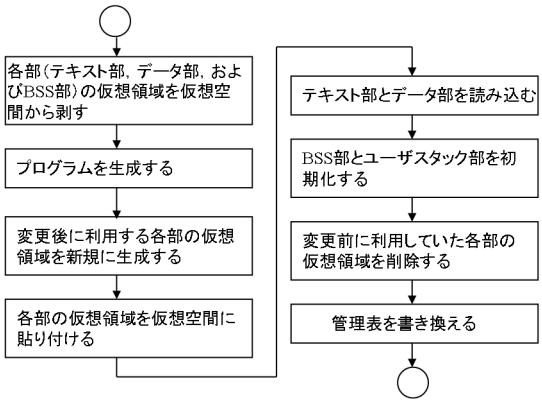

図 4 従来の実行プログラムの変更処理

プログラムを再利用する．テキスト部の仮想領域は，共有可能な仮想領域を検索し，共有可能な仮想領域が存在する場合には，共有可能な仮想領域を仮想空間に貼りつける．また，共有可能な場合は，テキスト部の内容も利用もできるため，テキスト部の内容を読み込まなくてよい．共有可能なテキスト部の仮想領域が存在しない場合は，再利用可能な仮想領域が存在するかを確認し，再利用可能な仮想領域がある場合には，その仮想領域を仮想空間に貼り付けて再利用する．データ部と BSS 部の仮想領域については，共に，再利用可能な仮想領域が存在するかを確認し，再利用可能な仮想領域がある場合には，その仮想領域を仮想空間に貼り付けて再利用する．

また，従来の手法では変更前に利用しているテキスト部，データ部，および BSS 部の仮想領域を削除していた．そこで，これらを削除せずに，再利用可能資源管理表に登録するように変更した．ただし，他のプロセスとテキスト部を共有している場合には，他のプロセスがテキスト部の仮想領域を利用しているため，テキスト部の仮想領域を再利用可能なように再利用可能資源管理表に登録しない．

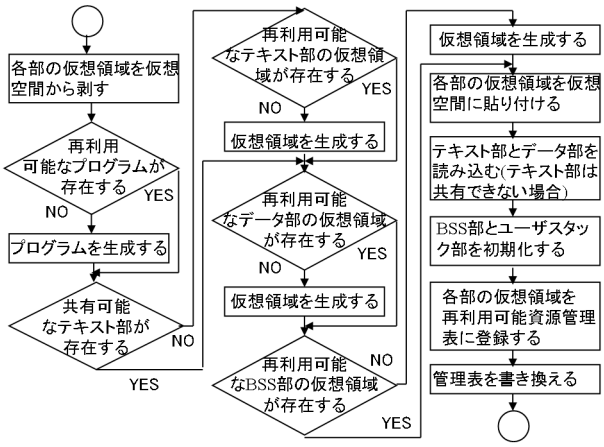

図 5 資源の再利用を行なう実行プログラムの変更処理

3.2 開始位置の変更における資源の再利用

開始位置の変更の様子を図 6 に示す．初期状態に変更する際は，データ部，BSS 部，ユーザスタック部，およびレジスタ群の内容を初期化する．指定された位置に変更する際は，当該の指定された位置における情報が必要となる．この情報は，プロセスが利用するデータ部，BSS 部，およびユーザスタック部の内容と，レジスタ群の情報からなる．指定され

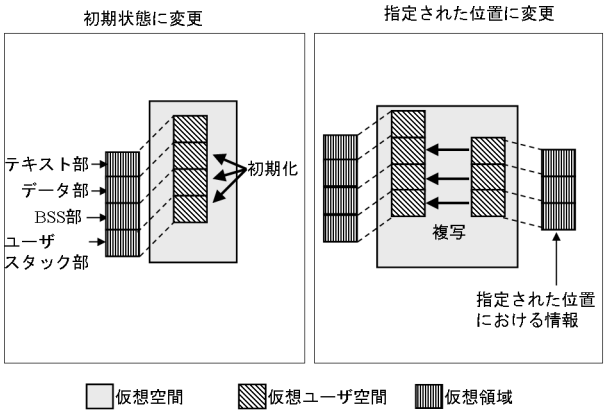

図 6 開始位置の変更の様子

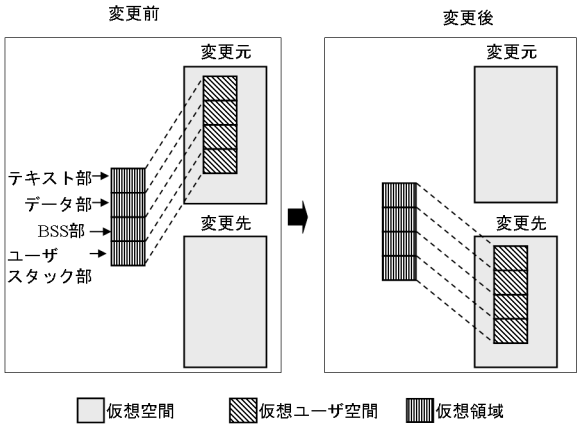

図 7 動作空間の変更の様子

た位置における情報をデータ部，BSS 部，ユーザスタック部，およびレジスタ群へ複写することで，指定された位置に変更する．

開始位置の変更では，初期状態への変更，指定された位置への変更，共に，新規にプロセス構成資源を生成する処理や，削除する処理がない．よって，開始位置の変更では，再利用可能な資源や，再利用可能資源管理表に登録する資源はない．

3.3 動作空間の変更における資源の再利用

同一計算機内の別空間に変更する様子を図 7 に示す．同一計算機内の別空間に変更するには，変更元の仮想空間から当該の仮想領域を剥し，その仮想領域を変更先の仮想空間へ貼り付けることで仮想領域の貼り変えを行なう．従来の手法では，図 8 の (a) に示すように，仮想領域を変更先の仮想空間に貼り付ける際に，新規に仮想空間を生成している．これに対し，動作空間の変更における資源の再利用として，図 8 の (b) に示すように，以下の方法で仮想空間を再利用するようにした．

（1）既存の仮想空間の資源識別子を指定して仮想空間を利用する

（2）既存の仮想空間の資源識別子を指定しない場合は，再利用可能な仮想空間の検索を行ない，再利用可能な仮想空間があれば，その仮想空間を再利用する

これらの処理により，変更先の仮想空間を新規に生成する処理を，既存の仮想空間の利用や，再利用可能な仮想空間の再利用を行なう処理に置き換えることができる．

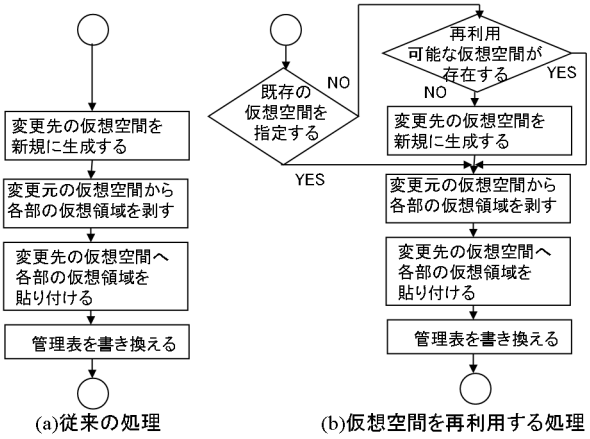

図 8 動作空間の変更処理

他計算機の空間に変更する処理は，まず始めに，変更先の計算機の空間上に，カーネルプロセスを生成する．次にカーネルプロセスの実行プログラム，開始位置，および動作空間をプロセス変身機能を利用して，動作空間を変更するプロセスのものに変更する．そして，変更元のプロセスを削除する．このように，プロセスを他計算機上の空間に変更する処理には，新規にプロセス構成資源を生成する処理やプロセス構成資源を削除する処理はない．このため，プロセスを他の計算機上に変更する機能には再利用可能な資源や，再利用可能資源管理表に登録する資源はない．

4 評価・考察

Tender に実現したプロセス変身機能における資源の再利用の効果を評価するために，実測による評価を行なった．測定を行なった処理対象は，実行プロ

グラムの変更と動作空間の変更において，各再利用可能な資源を再利用した場合の処理時間である．測定では各処理を呼び出してから戻ってくるまでの時間を，ハードウェアカウンタにより計測した．実測には Pentium Ⅲ 800MHz の計算機を用いた．

4.1 実行プログラムの変更処理時間

実行プログラムの変更における各再利用可能な資源を再利用した場合の効果について述べる．資源の再利用の効果を評価するために実測した条件として，実行プログラムの変更前のテキスト部の大きさ，データ部の大きさ，および BSS 部の大きさを各々4KB に固定した．実行プログラムの変更後のプログラムについてはデータ部の大きさと BSS 部の大きさを 4KB に固定し，テキスト部の大きさを可変にした．なお，BSS 部の仮想領域を再利用した場合の効果は，データ部を再利用した場合の効果と同様な結果であった．また，BSS 部を削除せずに再利用可能資源管理表に登録した場合の効果は，データ部を削除せずに再利用可能資源管理表に登録した場合の効果と同様な結果であった．

4.1.1 プログラムの再利用

実行プログラムの変更において，プログラムを再利用した場合の処理時間を図 9 に示す．プログラムを再利用した場合の実行プログラムの変更にかかる処理時間と，資源を再利用しない場合の実行プログラムの変更にかかる処理時間の差は，変更後のテキスト部の大きさが 4KB や 8KB の場合には，ほとんどない．プログラムの再利用による効果は低いといえる．これは，プログラムの生成はプレートを利用して，ディスク I/O が発生しない条件で測定したためである．

4.1.2 テキスト部の再利用

実行プログラムの変更において，テキスト部の仮想領域を再利用した場合の処理時間を図 10 に示す．テキスト部の仮想領域を再利用した場合の実行プログラムの変更にかかる処理時間は，資源を再利用しない場合の実行プログラムの変更にかかる処理時間に比べ，変更後のテキスト部の大きさが 4KB の場合には約 28 マイクロ秒短く，変更後のテキスト部の大きさが 32KB の場合は約 198 マイクロ秒短い．このように，テキスト部の仮想領域を再利用した場合の処理時間と資源を再利用しない場合の処理時間の差はテキスト部の大きさが大きくなるにつれて広がっている．これは，テキスト部の仮想領域を再利用した場合の実行プログラムの変更は，確保する領域の大きさに比例して生成処理時間が増大する仮想領域 [2] を生成しなくてもよいためである．

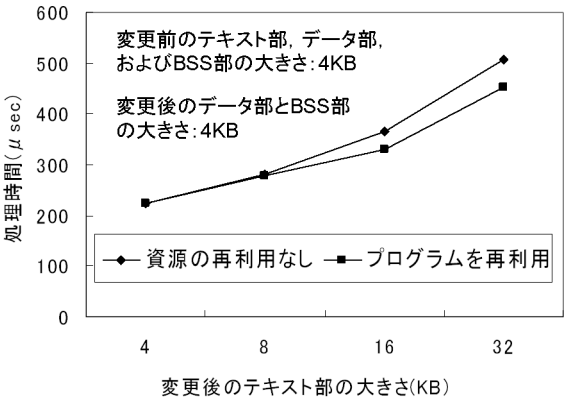

図 9 実行プログラムの変更におけるプログラムの再利用

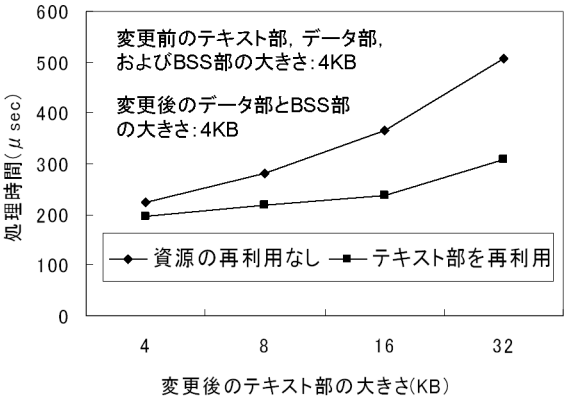

図 10 実行プログラムの変更におけるテキスト部の再利用

4.1.3 テキスト部の共有

実行プログラムの変更において，テキスト部を共有した場合の処理時間を図 11 に示す．資源を再利用しない場合のプログラムの変更にかかる処理時間は，テキスト部の大きさが大きくなるにつれて増大しているのに対して，テキスト部を共有した場合の実行プログラムの変更にかかる処理時間は，テキスト部の大きさによらずほぼ一定である．これは，テキスト部を共有する場合の実行プログラムの変更の処理では，新たにテキスト部の仮想領域を生成しないでよいだけでなく，テキスト部の内容も利用するため，テキスト部の内容を仮想ユーザ空間に読み込まなくてもよいためである．

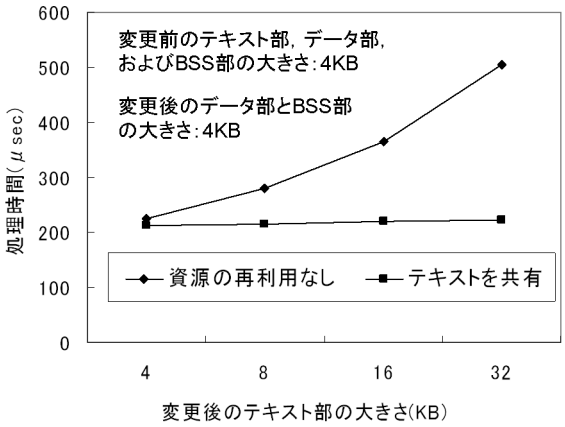

図 11　実行プログラムの変更におけるテキスト部の共有

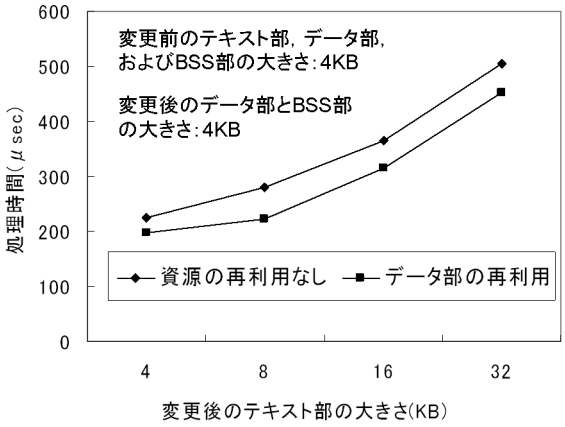

図 12　実行プログラムの変更におけるデータ部の再利用

4.1.4　データ部の再利用

実行プログラムの変更において，データ部の仮想領域を再利用した場合の処理時間を図 12 に示す．データ部の仮想領域を再利用した場合の実行プログラムの変更にかかる処理時間は，資源を再利用しない場合の実行プログラムの変更にかかる処理時間に比べて，変更後のテキスト部の大きさが 4KB の場合は約 28 マイクロ秒短いが，変更後のテキスト部の大きさが 8KB 以降は約 54 マイクロ秒短くほぼ一定である．これはデータ部の仮想領域を再利用する場合の実行プログラムの変更は，変更後のデータ部の仮想領域を作成しなくてもよいためであり，実測した条件ではデータ部の大きさはすべて同じ大きさであるため，両者の処理時間の差がほぼ一定である．

4.1.5　テキスト部の保存

実行プログラムの変更において，テキスト部の仮想領域を削除せず，仮想領域を残した場合の処理時間を図 13 に示す．テキスト部の仮想領域を削除せず，再利用可能資源管理表に登録した場合の実行プログラムの変更の処理時間と資源を再利用しない場合の実行プログラムの変更の処理時間の差はわずかである．したがって，実行プログラムの変更において，テキスト部の仮想領域を削除せず，再利用可能資源管理表に登録した場合の効果は，他の資源の再利用と比べて低いといえる．しかし，別に示したように，仮想領域を削除しないことにより，他のプロセスがこの仮想領域を共有したり，再利用する場合の効果は大きい．

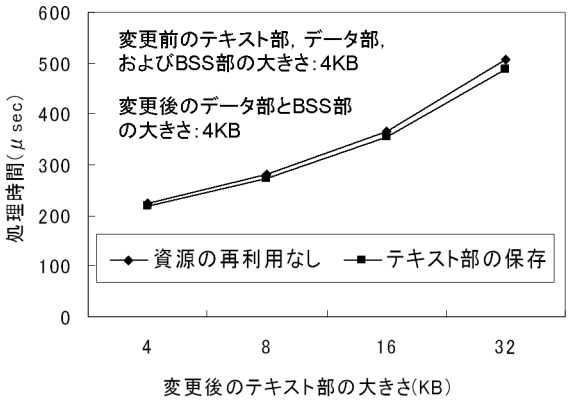

図 13　実行プログラムの変更におけるテキスト部の保存

4.1.6　データ部の保存

実行プログラムの変更において，変更前のデータ部の仮想領域を削除せず，再利用可能資源管理表に登録した場合の処理時間を図 14 に示す．データ部の仮想領域を削除せず，再利用可能資源可能管理表に登録する場合の実行プログラムの変更の処理時間と資源を再利用しない場合の実行プログラムの変更の処理時間の差はわずかである．実行プログラムの変更において，データ部の仮想領域を再利用可能資源可能管理表に登録する場合の効果は他の資源の再利用と比べて低いといえる．しかし，別に示したように，この仮想領域を削除しないことにより，他のプロセスがこの仮想領域を再利用する場合の効果は大きい．

4.2　動作空間の変更処理時間

動作空間を変更するプログラムのデータ部と BSS 部の大きさを 4KB に固定し，テキスト部の大きさを可変にした場合の，同一計算機内での動作空間の変

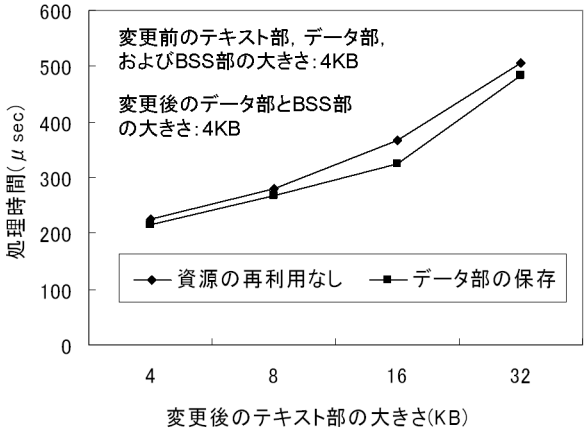

図 14 実行プログラムの変更におけるデータ部の保存

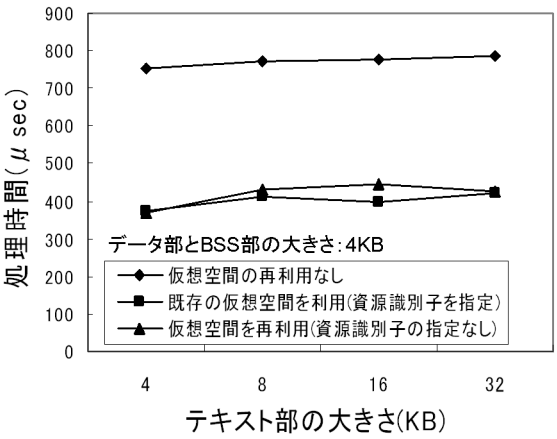

図 15 動作空間の変更における仮想空間の再利用

更処理時間を図 15 に示す．

既存の仮想領域を利用する場合や仮想空間を再利用する場合は，仮想空間を再利用しない場合に比べてその処理時間が約 380 マイクロ秒短く，約半分である．この差は，前者二つの場合は仮想空間を新規に生成しないためである．したがって，動作空間の変更における仮想空間の再利用効果は大きいといえる．

また，既存の仮想空間を利用する場合と，仮想空間を再利用する場合の動作空間の変更処理時間はほぼ同じである．二つの違いは，再利用可能な仮想空間の検索を行なうのにかかる時間である．文献 [2] により，再利用可能な仮想空間は O（1）の時間で検索を行なうことができるので，両者の処理時間の差はほとんどない．

そして，既存の仮想空間の利用や仮想空間の再利用による効果はテキスト部の大きさによらず一定である．これは仮想空間を生成する時間はテキスト部の大きさによらず一定だからである．

5 おわりに

プロセスの実行環境を変更する機能であるプロセス変身機能の高速化として，資源の再利用効果について述べた．プロセス変身機能の一つである実行プログラムの変更の資源の再利用として，変更後に利用するプログラム，テキスト部の仮想領域，データ部の仮想領域，および BSS 部の仮想領域を再利用することを実現した．また，テキスト部については共有可能な仮想領域があれば，共有することにより，仮想領域を新規に作成する処理とテキスト部の内容を仮想ユーザ空間に読み込む処理を削減した．そして，変更前に利用していたテキスト部の仮想領域，データ部の仮想領域，および BSS 部の仮想領域を削除していた処理を，削除せず再利用可能資源管理表に登録することを実現した．同一計算機内での動作空間の変更における資源の再利用として，変更先の仮想空間に仮想領域を貼り変える際に既存の仮想空間や再利用可能な仮想空間を再利用することを実現した．

プロセス変身機能における資源の再利用による効果を明確化にするために，実測による評価を行なった．実行プログラムの変更では，仮想領域の再利用による効果が大きく，特にテキスト部を共有した場合の効果が大きいことを示した．動作空間の変更では，既存の仮想空間の利用や，仮想空間を再利用することで，資源を再利用しない場合に比べて処理時間が約半分になることを示した．

また，開始位置の変更と他計算機上の空間に移動する機能には再利用可能な資源がないことを述べた．

残された課題としてプロセス変身機能を応用プログラムで利用した時の性能評価がある．

参考文献

[1] 石井陽介, 谷口 秀夫: “位置透過に利用可能な構成要素を用いたプロセス変身機能”, 情報処理学会論文誌,Vol.44, No.7,pp.1666-1679(2003).

[2] 田端 利宏 , 谷口 秀夫: “プロセス構成資源の効率的な再利用を目指した資源管理法の提案”, 情報処理学会論文誌：コンピューティングシステム, Vol.44, No.SIG10(ACS2), pp.48-61 (2003).