B_030

*AnT*における高速なプロセス間通信の実現

Realization of High Speed Inter-process Communication for *AnT*

梅本 昌典 Masanori Umemoto　田端 利宏 Toshihiro Tabata　乃村 能成 Yoshinari Nomura　谷口 秀夫 Hideo Taniguchi

1．はじめに

我々は，新しいハードウェアやサービスへの適応制御機能を持つ*AnT* オペレーティングシステム[1](以下*AnT*と呼ぶ)を開発している．*AnT* はマイクロカーネルの構造を持つ．マイクロカーネルでは，頻繁に発生するプロセス間通信のデータ授受でのオーバヘッドが問題になっており，プロセス間通信を高速化することが重要である．そこで，*AnT* ではコア間通信データ域(ICA:Inter-core Communication Area)と名付けた領域を導入することで，ゼロコピーによるプロセス間通信を実現し，プロセス間通信処理を高速化した．本稿では，*AnT*におけるプロセス間通信の実現方式について述べる．

2. プロセス間通信

*AnT*では，ICA と名付けたプロセス間でのデータ授受用の領域を介して，プロセス間通信を行う．ICA のアドレス空間は，すべてのユーザプロセスの仮想空間で予約され，プロセス毎にアクセス権のあるページのみがマッピングされる[2]．以下にプロセス間通信の手順を示す．

(1) 送信側プロセスは自分の仮想空間の ICA に必要量の通信域を作成する．
(2) 作成した通信域にデータを書き込む．
(3) コア呼び出し機能を用いて，受信側プロセスにデータを送信する．
(4) 受信側プロセスは，コア呼び出し機能によって，データの到着と ICA 上での通信域の貼り付けアドレスを知ることができる．

ICA 上の通信域を介したデータの送受信は，送受信プロセスの仮想空間のマッピング表を書き換えることにより実現する．これをページの貼り替えと名付ける．

ICA を利用した高速なプロセス間通信を実現するには，以下の課題がある．

(1) 高速なページの貼り替え処理
(2) ICA の外部断片化の防止

3. コア間通信データ域の実現方法

3.1 論実アドレス変換処理の高速化

AnT のゼロコピーによるプロセス間通信において，高速なプロセス間通信を実現するには，ページの貼り替え処理を高速化することが必要である．ページの貼り替え処理は，貼り付けと剥がしの処理からなる．ページの貼り付け処理では，送信先プロセスのマッピング表の更新時に送信元プロセスのマッピング表を参照し，貼り替え対象ページの物理アドレスを取得してから，更新処理を行う必要がある．マッピング表は 2 段の変換表になっており，その参照オーバヘッドが貼り付け処理において比較的大きいと推察できる．そこで，OS の開始処理において，ICA 用の実メモリ領域をまとめて確保し，仮想空間の論理アドレスとの関係を一意に決定することにした．その様子を図 1 に示す．

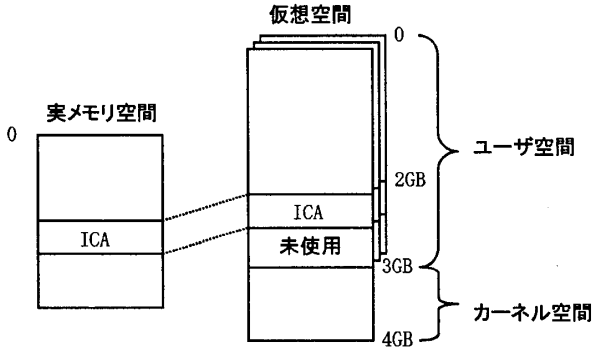

図1　ICAと仮想空間の関係

図 1 のように，論理アドレス 2GB から 3GB は ICA 用に予約されており，物理アドレスと論理アドレスは 1 対 1 に対応している．これにより，ICA における論実アドレス変換を以下の式により，簡単に行うことができ，ICA のページ貼り付け時の送信プロセスのマッピング表の参照回数を削減することができる．

（物理アドレス）　＝　（論理アドレス）－（2GB）
＋（ICA の先頭物理アドレス）

また，貼り替えでは，同じ論理アドレスへ通信域を貼り付ける．これにより，ゼロコピーによるプロセス間通信を実現できる

ただし，利用可能な ICA の大きさは，OS 起動時に決定されるため，その大きさを適切に指定し，効率的に利用する必要がある．

3.2 外部断片化の防止

ICA の大きさは，OS 起動時に固定であり，外部断片化が起こらないように ICA を利用する必要がある．外部断片化は，ICA の利用時間が異なるプロセスが，ICA の確保と解放を繰り返すことで発生する．このような状態を避けるため，ICA 確保時に flag を導入した．flag は，利用する分の ICA が一時的かどうかを示す．一時的に利用する領域は，ICA の先頭から，長期的に利用する領域は，ICA の後方から利用する分の ICA を確保することとした．これにより，外部断片化の発生を防ぎ，効率良く ICA を利用できるようにする．

岡山大学大学院自然科学研究科

表1　ICA操作に関する提供インタフェースと提供先

形式	概要	提供先モジュール		
		外コアとプロセス	内コア	メモリ領域管理の内部モジュール
(1) createica(size, flag)	size分の領域をflagで指定された方法でICAに作成する	○	○	○
(2) deleteica(vaddr)	vaddrから始まる領域をICAから削除する	○	○	○
(3) attachica(vmid, vaddr, prot)	vmidとvaddrで指定した領域を，protで指定した保護情報でICAに貼り付ける	—	○	○
(4) detachica(vmid, vaddr)	vmid，vaddrで指定された領域をICAから剥がす	—	○	○
(5) getmem(op, size, flag)	size分のICA内の空き領域をflagで指定された方法で確保する	—	—	○
(6) freemem(raddr)	raddrで指定されたICA内の領域を解放する	—	—	○

4. 提供インタフェース

実装したICA操作に関する提供インタフェースと提供先を表1に示す．作成，削除，貼り付け，剥がし，確保，および解放の計6つのインタフェースを提供する．ここで，これらのインタフェースの提供先モジュールとして，***AnT*** のOSの一部である外コアとプロセス，内コア，および内コアの一部であるメモリ領域管理の内部モジュールの3つに分類した．プロセス間通信を行うため，外コアとプロセスには領域の作成や削除を要求できるインタフェースが必要である．ページの貼り替えは，内コアによって行われる．このため，内コアにはマッピング表を変更できるインタフェースが必要である． ICAの各ページに対する未使用や使用中の管理は，メモリ領域管理の内部モジュールによって排他的に行う必要がある．このため，確保と解放はメモリ領域管理の内部モジュールにのみ提供している．

なお，freemem()において，解放する大きさはICAに領域を作成した時の領域の大きさに等しいこととした．したがって，解放する領域の大きさをICAの管理表から得るように処理する．

5. 評価

5.1 評価項目と測定環境

ICAを利用したプロセス間通信の高速性を評価するため，プロセス間通信処理の主要な処理時間である貼り付け時間について評価する．具体的には，ICAの空間への貼り付け処理時間と4KBページ域の空間への貼り付け処理時間を比較する．4KBページ域とは，4KB境界を越えると論実アドレスの連続が保証されない領域である．したがって，この両者の処理の大きな違いは論実アドレス変換処理である．

測定は，Pentium4 2.4GHzのPC/AT互換機上で行った．

5.2 考察

貼り付けに要する処理時間を図2に示す．図2から，貼り付けるデータサイズに対する処理時間の増加量は

(1) ICAの場合；約0.058μsec/4KB

(2) 4KBページ域の場合；約0.124μsec/4KB

である．したがって，ICAを利用したプロセス間通信では，約0.066μsec/4KBの高速化が見込める．

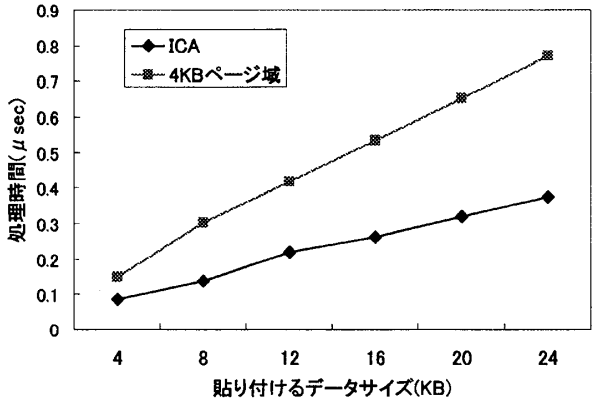

図2　貼り付けに要する処理時間

6. おわりに

AnT におけるプロセス間通信方式として，ゼロコピーによる高速な方式を示した．具体的には，論実アドレス変換を高速化できるICAを実現し，プロセス間通信は，ICA内のデータ授受領域を空間間で貼り替えることで行う．これにより高速なプロセス間通信が可能になる．また，評価により、プロセス間通信処理の主要な処理である貼り付け処理の時間を4KBページ域に比べ，約0.066μsec/4KB高速化できることを明らかにした．

残された課題として，プロセス間通信機能を用いた評価がある．

本研究の一部は，科学研究補助金 基盤研究（B）「適応性と頑健性を有する基盤ソフトウェアのカーネル開発」（課題番号:18300010）による．

参考文献

[1] 谷口 秀夫，乃村 能成，田端 利宏，“***AnT*** オペレーティングシステムの設計”，情報処理学会第68回全国大会講演論文集，分冊1，pp.41-42(2006)

[2] 田端 利宏，梅本 昌典，安達 俊光，谷口 秀夫，“***AnT*** オペレーティングシステムのメモリ領域管理”，情報処理学会第68回全国大会講演論文集，分冊1，pp.45-46(2006)