

1H-4

# 分散共有メモリを用いた 実時間での記憶管理に関する考察\*

河野 通宗 飯田 浩二 矢向 高弘 安西 祐一郎†

慶應義塾大学‡

## 1 はじめに

マルチプロセッサを機能別に割り当てる分散制御システムにおいて、仮想記憶機構に実時間制約の保証が可能なページング機構を付加することは難しい。

本研究では、分散共有メモリをページングの補助記憶に用い、ページング時間が予測可能な仮想記憶機構の実装を試みる。

## 2 実時間での仮想記憶

デマンドページングによる仮想記憶機構を持つOSでのページングは、バスエラーが起きてからのハンドリングによるエラーハンドリングを基に実装されている[1]。一方リアルタイムOSにおいては、ページングは非周期的な割り込みタスクという形で実装し得る。スケジューラはこのタスクの到着を受けて再スケジューリングを行うが、ページングに要する時間が予測不可能なために周期的タスクがデッドラインミスを起こす可能性が生じる。

これを解決するには、メモリ確保の際のページングの発生をなくし、タスクがページング待ち状態になるのを避けるのが一番の方法である。そのためにはタスクの最大確保領域を実行前に特定でき、かつタスクの実行中そのサイズ分の実メモリを割り当て続けなければならない。そのため既存のリアルタイムOSでは、ページングをせずに実メモリ空間のみの使用に限定している[4]。

しかし、実行させるタスクをユーザが後から変更できるようなシステムの場合、予め十分なメモリを用意することは難しい。そこでリアルタイム制約のないタスクのメモリ空間だけをページングの対象とすることを考える。そのページングに要する時間を予測可能にできれば、ページングのために周期的タスクのデッドラインミスを避けることが可能となる。そこで、

- ページングを、実行時間が予測可能な非周期的タスクとしてスケジューリングする

ことを考える。これはバスで結合されている分散共有メモリがページングデバイスで、アクセス時間の予測が可能であれば実現可能であると考えられる。そこで

分散共有メモリをページングデバイスとし、時間制約下でのページングの実装について検討し、実験・評価する。

## 3 設計

仮想アドレス空間の生成・管理、およびページング実行部分に関して

- タスク毎にメモリ空間を独立させる
- メモリの確保の際にページングを起こす事のないよう、出来るだけ適当なページサイズ分を常に空き領域として確保しておく
- ページ毎に、ページアウトの禁止を指定出来る
- 管理スレッドはページ監視部とページング実行部から構成し、別個に容易に変更出来るようにするという方針の下に設計を行った。

### 管理スレッドの構成

$\mu$ -PULSER[2]への仮想記憶の実装およびページング処理実現のために、以下の3つの管理スレッドを設計した。スレッドを機能毎に分けることにより、ページングアルゴリズムのみの変更が、スレッドを置き換えるだけで実現可能である。

- **空きメモリ監視スレッド**  
メモリマネージャの管理するメモリテーブルを周期的に参照し、必要ならばページイン/アウトスレッドを起動する。また物理メモリテーブルの参照マークを書き換え、ページアウトの候補となるページにマークをつける。
- **ページイン/アウト実行スレッド**  
空きメモリ監視スレッドにより起動される非周期スレッドである。ページインとページアウトで別々のスレッドに分けている。共有メモリへの実際のアクセスを受け持つ。

## 4 実装

### 4.1 ASPIRE アーキテクチャ

実装対象であるロボットアーキテクチャASPIREはパーソナルロボットを目的としたアーキテクチャで、センサ、モータなどの各要素がモジュールに分割され、メインモジュールがそれら全体を統括する[3]。各モジュールにはプロセッサ、ローカルメモリ、共有メモリが実

\*Real-time memory management with distributed shared memory

†Michimune Kohno, Kouji Iida, Takahiro Yakoh, and Yuichiro Anzai

‡Keio University

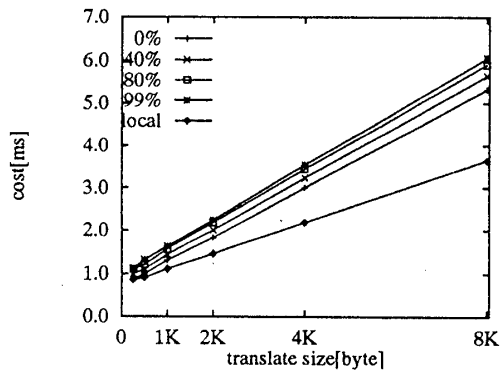


図 1: 共有メモリへのメモリ転送時間

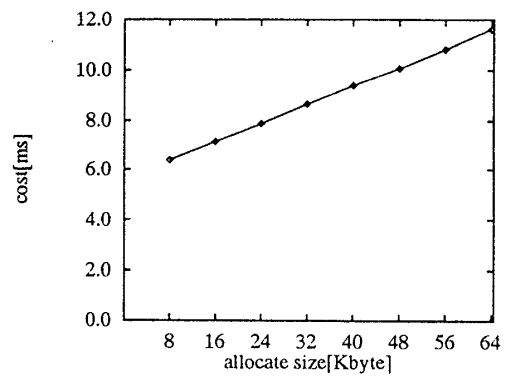


図 2: ページ確保のコスト

装されていて、VMEバスを介しての共有メモリの相互参照が可能である。

現在、メインモジュールにはMC68030(25MHz)をプロセッサに持つAVME-130-1(RAM 8MB, ROM 2MB)を採用している。モータモジュールおよびセンサモジュールは、我々の研究室で設計された、TMP68301を用いたモジュールである。ROM, ローカルRAM, 共有RAMを1MBずつ実装している。

## 4.2 実装

仮想記憶機構は、メインモジュールのプロセッサに内蔵されているMMUを用いて実装した。ページアウトされるページの選択には疑似LRUアルゴリズムを使用した。ページサイズは8KBにした。ページイン/アウトの起動されるスレッシュホールドはそれぞれ256KB, 128KBに設定し、頻繁なページング動作が起きないようにしている。

## 5 評価

### 5.1 共有メモリ間の転送

共有メモリへのメモリ転送を行うスレッドの実行時間を図1に示す。この図はバスの使用率を変えて測定したものだが、ほぼ線形の転送時間が得られている。DMA転送を頻繁に行う環境下ではこの線形性が崩れると予想されるが、機能毎に分かれた分散環境ではバスの使用率が極度に高い状態が長く続くことはそれほど多くないと考えられる。ASPIREでは、タスクの実行時に共有メモリのわずかな部分が直接参照されるのみである。従って、機能分散されたバス結合型分散モデルではページングに要する時間が予測可能であると考えられる。

### 5.2 測定

図2に、リアルタイムスケジューラの下での、ページングを伴わないページ確保のコストを示す。図から、ページ確保の時間はほぼ完全に線形であり、ページングの起きない状態でのメモリ確保の時間予測は十分可能であると言える。

## 6 まとめ

本稿では、実行時間が予測可能な仮想記憶機構の $\mu$ -PULSERへの実装について、その実装および評価を示した。ASPIREアーキテクチャでの、共有メモリのアクセス時間が予測可能である事を示し、ページング用のスレッドを細かく分ける事により、ページングアルゴリズムの変更が容易に行えることを述べた。

評価としてメモリ確保に要する時間を示し、ページングの起きない状態でのメモリ確保、および共有メモリとのページングにかかる時間が予測可能であることを示した。

今後はこの結果を用いて、ページングの起きる状況でのメモリ確保の実時間性の保証について検討する。

## 参考文献

- [1] David L.Black. *Scheduling and Resource Management Techniques for Multiprocessors*. CMU-CS-90-152, Jul. 1990.
- [2] 矢向高弘, 菅原 智義, 安西 祐一郎. “ $\mu$ -PULSER: パーソナルロボットを構築するためのオペレーティングシステム”, 電子情報通信学会論文誌, Feb. 1994.
- [3] 山崎 信行. “パーソナルロボット用アーキテクチャ ASPIRE の設計と実装”, 慶應義塾大学 修士論文, 1993.
- [4] 徳田 英幸, “分散リアルタイム OS の技術動向”. コンピュータソフトウェア, Vol 9, No. 3. pp.182-192, May 1992.