

超並列システムのための大域的仮想仮想記憶(GVVM)の実装設計

3F-8

伊藤 民哉 平野 聰⁺⁺電子技術総合研究所

1はじめに

超並列システムのための超流動OS[1]の大域的仮想仮想記憶(GVVM)[2]をIntel Paragon XP/S上に実装している。本論文では、Machの外部ページャを利用してGVVM処理系の実装設計の方針について述べる。

2 設計方針

GVVMでは、あるPEにおいてページアウトが発生した場合に、直ちに2次記憶に待避させるのではなく、他のPE上により使用頻度の低い物理ページが存在した場合には、そこをページアウト先として使用する。ページアウトされるべきページより使用頻度の低いページが見つからないときに初めて2次記憶にページアウトされる。これにより、PE空間全体に渡った物理メモリ資源の有効利用を図ることを目的としている。

本実装では、移植性と実用上の性能の両立を図ることを目的とする。そこで、GVVMをMachの外部ページャ(以下、GVVMページャとする)として実装し、高速化のためGVVMページャ同士の通信には、Machのタスク間通信プリミティブであるポートは利用せず、Paragon固有の通信プリミティブを用いることにした。

3 処理系の全体構成

図1にPE内でのGVVMページャの構成を示す。システム内の全てのPEで同様の構成をとる。

各PEに配置されたGVVMページャとMachカーネルとの間では、EMM(External Memory Manager)インターフェースに従って、ポートを介してメッセージ通信が行われる。アプリケーションタスクは、メモリオブジェクトを取得するために一度だけGVVMページャと直接通信するが、カーネルによりそのメモリオブジェクトを自分自身の仮想メモリ空間にマップされた後は、GVVMページャと直接通信することはない。アプリケーションタスク内で発生したページフォルトは、カーネルによって検出され、GVVMページャにメッセージが通知される。

GVVMページャ内では、メモリオブジェクト毎に独立した仮想メモリ管理スレッドが起動される。カーネルから発行されるメモリオブジェクトに対する各種のメッセージは、このスレッドが受信する。受けたメッセージの内容により、ページインやページアウト等の処理が行われる。

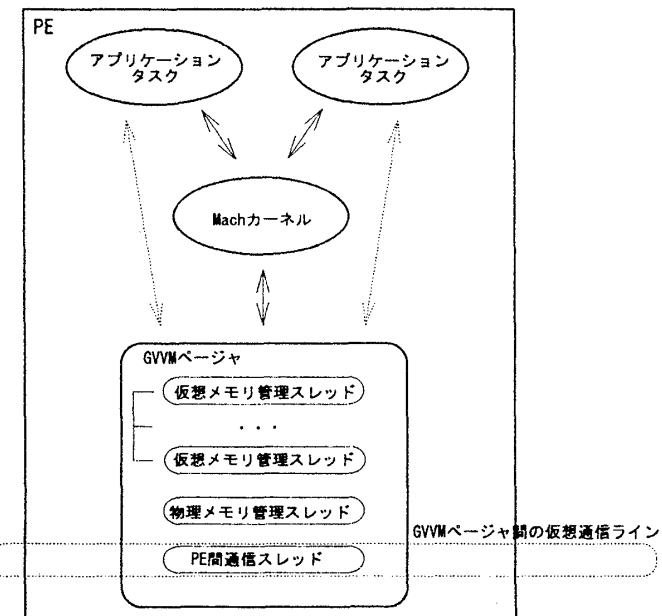


図1 PE内の構成

これらの処理を自PE内で解決できない場合、例えば、他のPEにページアウトされているページをページインする場合は、PE間通信スレッドを通して相手側PEに要求を送る。

一方、物理メモリ管理スレッドと、PE間通信スレッドはGVVMページャあたり1つだけ存在する。物理メモリ管理スレッドは、物理メモリのページ使用頻度の測定とPE間での物理ページの融通を行う。

PE間通信スレッドは、Paragonの通信プリミティブを用いてスレッド同士で直接通信を行うことにより、GVVMページャ間の通信を提供する。

4 メモリ管理

GVVMページャでは、物理メモリの使用頻度を把握していかなければならないが、Machでは物理メモリおよびその管理情報は、ユーザタスクであるGVVMページャに公開されていない。そこで、物理メモリおよびその管理テーブル等をGVVMページャ内でシミュレートする必要がある。このため、以下のデータ構造を用意した。

仮想物理メモリ Machの仮想記憶サブシステムを用いて、GVVMページャ内に仮想的に連続なメモ

リ空間をひとつ用意する。また、このメモリ空間のページと1対1に対応した管理情報マップを用意し、仮想物理メモリの状態を保持するものとする。

ページテーブル 生成されるメモリオブジェクト毎に用意される。PTE(Page Table Entry)には、メモリオブジェクトでのページが仮想物理メモリのどのページに割り付けられているか、または、どこにページアウトされているかの情報を記録する。以上のデータ構造の関係を図2に示す。

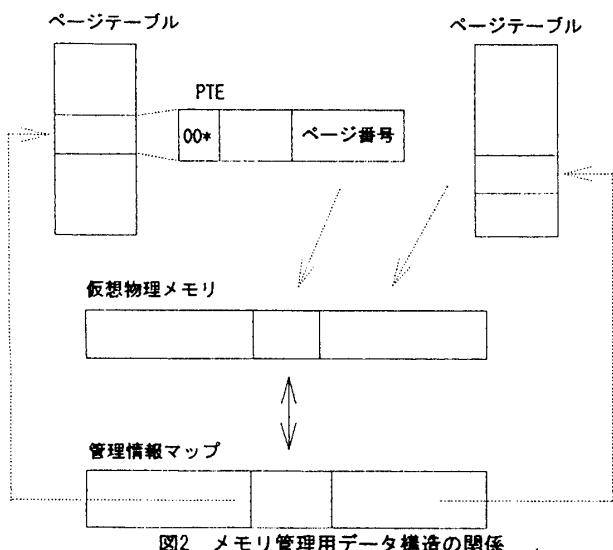


図2 メモリ管理用データ構造の関係

ここでは、メモリオブジェクトのページが、仮想物理メモリに在中している場合を示している。PTE中のページ番号は仮想物理メモリでのページ位置を示しており、このページと1対1に対応した管理情報マップの要素には、ページがどのページテーブルのPTEに対応しているかが示されている。他のPEにページアウト先として提供されているページの場合には、提供先に関する情報が代わりに置かれる。また、ページの使用頻度を示す情報や、ページのロック等に必要な情報もここに置かれる。管理情報マップの要素は使用頻度順のリストによって管理されており、使用頻度の低いページの取り出しが容易になっている。また、一定数のフリーページの確保とフリーページの有効な再利用を可能するために、フリーページに対応するマップの要素同士を双向リストで管理する。

5 処理系の実際の動作

sというノード番号を有するPE[s]でページフォルトが発生し、その解決のためにページインを行うが、フリーページの数が最低確保数を下回っているためにまずページアウトを行う必要が生じた場合を例に、GVVMページヤの実際の動作を説明する。

1. PE[s]上の仮想メモリ管理スレッド(以下、VT)

が、PTEを調べて対象ページがページアウトされている事を認識し、ページイン要請を物理メモリ管理スレッド(PT)に送る。

2. PTは、ページインを行うためにはページアウトが必要と判断する。
3. PTは、自PEのメモリ使用頻度を管理情報マップを基に求める。仮にその値をuとする。続いてスワップスペース探索戦略にしたがい、探索対象となるPEを複数選択し、各々への入札要求発行をPE間通信スレッド(CT)に依頼する。
4. CTを介して入札要求を受け取ったPEのPTは、自PEのメモリ使用頻度がuより小さいか調べて、そうであれば、入札に応じることを要求を発したPEに伝える。さもなければ、入札に参加しない旨を伝える。
5. PE[s]上のPTは、送られてきた入札の中から最も低い使用頻度を示したものを探用し、採用PE(PE[d]とする)に対しては、仮想ページアウトを発行し、他のPEには、入札失敗を送る。どこからも入札が入らなかった場合は、2次記憶にページアウトする。
6. PE[d]上のPTは、送られてきたページ内容をページバッファにいったん格納して、PE[s]のPTにページアウト処理終了を伝える。その後、提供場所に決まったページをページアウトするために、3からの処理が再帰的に行われる。このページアウトにより動かされるページの持ち主が他のPEの場合には、そのPTに対して、移動先が通知される。これを受け取ったPTは、PTEを変更する。

以上で、ページアウト処理が終了となる。この後PE[s]上のPTでページイン処理が開始される。

謝辞 本研究の一部はRWC計画の一環として「超並列システムアーキテクチャに関する研究」で行われたものである。関係各位に感謝する。

参考文献

- [1] 平野聰, 田沼均, 須崎有康, 濱崎陽一, 塚本享治. 超並列システム用オペレーティングシステム「超流動OS」の構想. 情報処理学会研究報告93-OS-58, pp. 17-24, 1993.
- [2] 平野聰, 田沼均, 須崎有康. 超並列システム用OS 「超流動OS」における大域的仮想仮想記憶. JSPP'93, pp. 237-244, 1993.