マイクロカーネル上の分散共有メモリサーバの構成

3T－4

斎藤 彰一　中村 素典　大久保 英嗣　大野 豊
立命館大学理工学部情報学科

1 はじめに

Mach[1] や Chorus[2] のように, マイクロカーネル (MK) と複数のシステムサーバによって構成されるオペレーティングシステム (OS) では, MK とシステムサーバ間のインタフェース (システムコール) をユーザに開放することで, ユーザが OS の機能を変更することが可能となっている. この代表的な例がページング機構を MK の外に配置することである. Mach では External Pager が, Chorus では Mapper がこれに相当する.

OS がユーザレベルのページング機構を提供することによって, ユーザは, アプリケーションのメモリアクセスパターンに適したページ置き換えアルゴリズムを実装することが可能になる. また, ページング用の 2 次記憶としてリモートマシンのメモリや 2 次記憶を使用することができる. さらに, これを応用することで, ユーザレベルで分散共有メモリ (Distributed Shared Memory) を実現することができる. 本論文では, 外部ページング機構を用いた分散共有メモリの実現について述べる.

分散共有メモリは, ネットワーク上の複数のプロセス間で共有される 1 つのアドレス空間であり, 各マシンのローカルのプロセス空間の一部にマッピングされて, 通常のアドレス空間と同様の方法でアクセスすることができる. ユーザレベルのページング機構を使用して, 我々が, 実現したユーザレベルシステムサーバを分散共有メモリサーバ (DSM サーバ) という.

本論文では, 外部ページング機構を利用した分散共有メモリの実現について検討を行い, 我々が Mach MK 上に実装した DSM サーバの構成について述べる.

A Structure of Distributed Shared Memory Server on Micro Kernel
Shoichi Saito, Motonori Nakamura, Eiji Okubo, and Yutaka Ohno
Department of Computer Science, Faculty of Science and Engineering, Ritsumeikan University
1916 Noji, Kusatsu, Shiga 525, Japan

2 ユーザレベルのページング

ユーザレベルのページングは, MMU を制御する仮想記憶管理部, ページング機構を備えたサーバ, さらにそれらの間のインタフェースによって構成される. Mach や Chorus では仮想記憶管理部は MK 内に実装されており, MMU の制御と仮想記憶の管理を行っている. また, ページング用サーバに対するインタフェースを提供している. 仮想記憶管理部では, ページフォルトが発生した場合に, 当該フォルトをページング用サーバへの要求に変換する. また, サーバからの要求によってページのマップ/アンマップを行う. サーバは, 仮想記憶管理部へページのマップ/アンマップを要求することで, ユーザ独自のページ置き換えを実現することができる. これらの機構は, Mach や Chorus では, システムサーバの中のファイルマネージャの実現に使用されている.

3 分散共有メモリの実現に必要な機構

分散共有メモリを実現するためには以下に示すような機構が必要となる. DSM サーバではこれらの機構を 3 つのマネージャによって実現している.

・分散環境でメモリイメージの共有を実現するためには, 各マシンにメモリイメージのコピーを持つ必要がある. そのために, それらのコピー間の一貫性制御が必要となる. また, 物理ページのマップ/アンマップも必要となる.

・MK ではファイルシステムを提供しない. 従って, ページング用 2 次記憶の入出力を独自に行う必要がある.

・分散環境で 1 つのメモリオブジェクトを共有するために, 共有メモリのマップや解放の管理を行う必要がある.

4 分散共有メモリサーバ

我々は, Mach の外部ページャ機構を利用して, Mach MK 上に DSM サーバの実装を行った. Mach における外部ページング機構は外部ページャと呼ばれている.

DSM サーバは外部ページャを使用した**分散共有メモリマネージャ(MM)**, 共有メモリオブジェクトの管理を行う**共有オブジェクトマネージャ(OM)**, ページング用 2 次記憶 (ページングファイルと呼ぶ) の管理を行う**ページングマネージャ(PM)** の 3 つのタスクで構成されている (図 1参照). 以下, それぞれについて説明する.

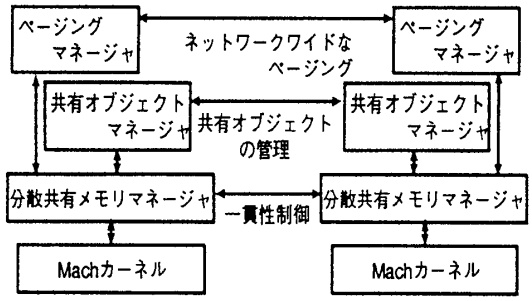

図 1: 分散共有メモリサーバ

・分散共有メモリマネージャ(MM)

ページング用サーバである. 主記憶上でページフォルトが発生した場合, カーネルからのメッセージに従ってメモリのページインとページアウトを行う. さらに, 各マシンの主記憶を分散共有メモリのキャッシュとして使用し, そのキャッシュに分散共有メモリのコピーの全体あるいは一部を配置する. この各マシンの共有メモリの内容の同一性を保証するために, 各マシンの MM との間で, 共有メモリのコピーに対して一貫性制御を行っている. 一貫性制御には, write-invalidate 方式を採用している. write-invalidate 方式では, 読み出しは同時に複数のマシンから可能であり, 書き込みは同時には唯一つのマシンでのみ可能である.

・共有オブジェクトマネージャ(OM)

共有メモリオブジェクトの管理を行う. 本来メモリオブジェクトはマシンローカルなものである. 本マネージャでは, これを各サイト間で協調しながら管理を行ない, ネットワークワイドな共有メモリオブジェクトとして扱うことを可能にしている.

・ページングマネージャ(PM)

ページング処理とそのための 2 次記憶の管理を行う. 本マネージャは Mach の UX サーバの i-node ページャを利用し, Unix ファイルへページングを行っている. DSM サーバの中で, UX サーバに強く依存する部分である. このため, 2 次記憶への入出力が発生する毎に, UX サーバへのシステムコールやコンテキスト切替が発生する. これを解決するためには, MK が提供する低レベルのカネールコールを用いて, UX サーバを使用することなく, 2 次記憶への入出力を行う必要がある. これにより, コンテキスト切り替えや, i-node 検索のコストを無くすことが可能となる. また, 本マネージャの特徴として, 各々のマシンが提供する 2 次記憶の領域をネットワーク上のすべてのマシンで共有している. これにより, 各マシンの 2 次記憶よりも大きな仮想空間を必要とする処理も行うことが可能となっている.

5 おわりに

本稿では, Mach の外部ページャを用いた DSM サーバの構成について述べた. Chorus が提供する外部ページング機構も, Mach のそれと同様の機構を提供していることから, 同様のシステムが構築可能と考えられる.

外部ページング機構に代表されるような, ユーザが OS のプログラムを変更することなく OS の機能の変更や拡張が行えるシステムを利用することで, 各々のアプリケーションに適した OS 環境を, ユーザ自身が構築することが可能になる. それによって, アプリケーションの性能向上が期待できる. また, 分散共有メモリのように, OS が提供していない機能も実現することができる.

参考文献

[1] Mike Accetta, Robert Baron, David Golub, Richard Rashid, Avadis Tevanian, and Michael Young: *Mach: A New Kernel Foundation for UNIX Development*, 1986 Summer USENIX Conference (1986).

[2] M. Rozier, V Abrossimov, F. Armand, I Boule, M Gien, M Guillemont, F. Herrmann, C. Kaiser, S. Langlois, P. Leonard, and W. Neuhauser: *Overview of the CHORUS Distributed Operating Systems*, Chorus Systemes Technical Report, CS-TR-90-25 (1990).