大規模アドレス空間内でのデータ保護方式の実現

4D−7

野末浩志、斎藤光男
(株) 東芝

1 はじめに

近年マイクロプロセッサの性能は大きく向上してきている。この中で、プロセッサのアドレス空間は、32bitから64bitへ移行しつつある [1]。64bitアドレス空間は、例えば、マルチメディアやオンライン・トランザクション処理などが持つ大規模なデータベースの処理へ応用されることが期待されている。

一方、マイクロプロセッサの性能向上は、分散処理環境の構築を可能にした。分散処理環境では、並列処理によって処理時間を短縮することが期待できるが、一般的にプロセス間通信を行なうカーネル処理のコストが大きい。このオーバヘッドを低減するために、ユーザ・レベルのプロセス間通信 [2] やユーザ・レベルでスケジューリングを行なう Light Weight Process [3] が研究されている。

我々は、64bitアドレス空間を利用して、単一仮想記憶方式に基づいたプログラム管理方式を検討している [5]。この方式では、クライアント／サーバ型プログラムにおいて、コンテクスト・スイッチに関わるオーバヘッドを削減できるという利点があるが、

- サーバのデータをクライアントの不正アクセスから保護することが困難である

という問題点がある。

そこで我々は、同一空間内でデータ隠蔽領域の作成を可能にする、新しいメモリ管理装置を提案する。

2 プログラミング・モデル

我々は、64bitアドレス空間を活かしたマイクロカーネル・レベルのOSを構想している。このOSのプログラム管理方式は、次の4つの特徴を持つ。

1. 単一仮想記憶方式

   ユーザ・レベルから各サーバ、OS、カーネルまで全てのプログラムを同一仮想アドレス空間内に配置する。これにより、コンテクスト・スイッチ回数を削減したり、スイッチに伴うキャッシュ整合性に関するオーバヘッド [4] を削減したりできる。

2. 一元化記憶方式

   データ・ファイルもプログラムと同じ仮想アドレス空間内に配置する。これにより、ファイル・アクセス時にカーネルの介入を回避したり、プロセッサのアドレッシング・モードを活用したりできる。

3. 空間とスレッドの分離

   カーネルは次の2つの資源を独立に扱う。

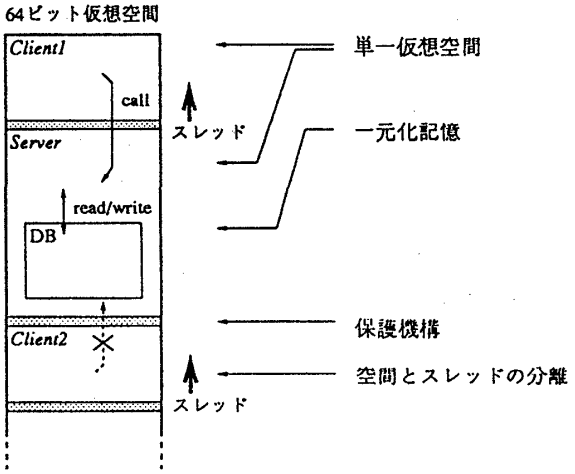

図 1: マイクロカーネルの概要

メモリ・セクション アクセスされるデータ、プログラムを格納する領域のこと。メモリが割り当てられ、ページングの対象となる。

スレッド アクセスするもの、すなわちプログラムを実行する主体。コンテクストが割り当てられ、スケジューリングの対象となる。

4. データ保護機構

   従来、ある仮想空間のデータやプログラムは、異なる仮想空間から直接アクセスできず、不正アクセスから保護されていた。本方式では、単一仮想空間方式を採るため、従来のメモリ管理機構では十分な保護が行なえない。そこで、仮想空間内で、データとプログラムを隠蔽 (encapsulate) して、他のメモリ・セクションからの不正アクセスを禁止するデータ保護機構を設ける。

3 データ保護方式

3.1 データ隠蔽の条件

不正アクセスを禁止するためには、次の3つの条件:

アクセス主体 一つの仮想空間内でも、スレッドよってアクセスして良いメモリ・セクションが異なる。従って、スレッドの身分を保証しなければならない。

アクセス起源 アクセスが許可されているスレッドであっても、正しいアクセス・ルーチンからのみアクセスしても良い。従って、アクセス・ルーチンのエントリポイントを保証しなければならない。

アクセス属性 データのRead/Write、命令fetchなどアクセスの仕方を示す。これは従来のアクセス制御と同様である。

が明らかでなければならない。

A Mechanism of Data Encapsulation in a Large Address Space.
Hiroshi NOZUE and Mitsuo SAITO.
Toshiba Corporation.

3.2 スレッドの身分の保証

メモリ・セクションへのアクセス許可は、アクセスを実行する主体、すなわちスレッド毎に与えられる。例えば、一般ユーザのスレッドは、システム領域へのアクセスが禁止される。またRPCなど通信を行なう場合、権限を与えられたスレッド同士のみがアクセスできるセクションが必要となる。一方、ライブラリのように不特定のスレッドからアクセスされるものもある。

このスレッドの識別は、メモリ・セクションに対してアクセスを許可するスレッドを指定することで行なう。スレッド毎にアクセスできる範囲が異なる様子を図2に示す。encapsulated領域は、あるエントリポイントを経由することでのみアクセス可能である領域を示す。

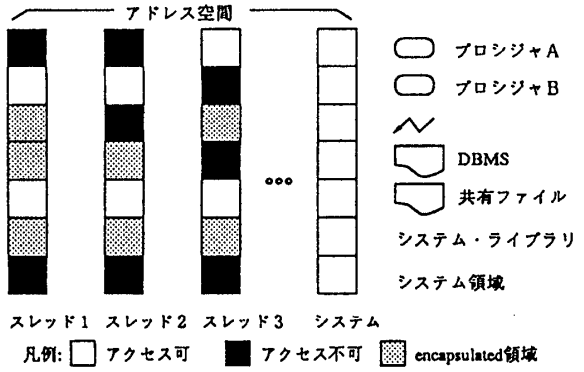

図 2: スレッドに関するアクセス制御

3.3 エントリポイントの保証

データが正しいアクセスルーチンを持つテキストからのみアクセスされる保証は、データ・セクションに対してアクセス元のセクションを指定することで行なえる。さらに命令が正しいアクセスルーチンを経由して実行していくように、次の概念を導入する。

ゲート・セクション ゲートは隠蔽された領域への入口となるメモリ・セクションである。隠蔽された領域のテキストは、このゲートから分岐された場合にのみ実行可能である。この識別は、メモリ・セクションに対して分岐元のセクションを指定することで行なう。(図3)

アクセス許可セクションの情報は、アクセス制御リストの中に格納される。例えば図3において、*Server* のGateセクションへは、自分自身(S-G)や *Client1* のTextセクション(C1-T)から分岐することが許可される。また、C1-Tセクションが直接 *Server*-Text内のルーチンへ侵入しようとしても、S-Tはゲート・セクションからの分岐以外許可しないので、不正アクセスが検出される。

3.4 高機能メモリ管理装置

本データ保護方式は、メモリ管理装置(MMU)の簡単な機能強化により実現可能である。

1. ページ・テーブルの構造(アクセス制御リスト)

   ページ毎に、従来のアドレス変換情報に加えて、アクセスを許可するアクセス主体とアクセス起源

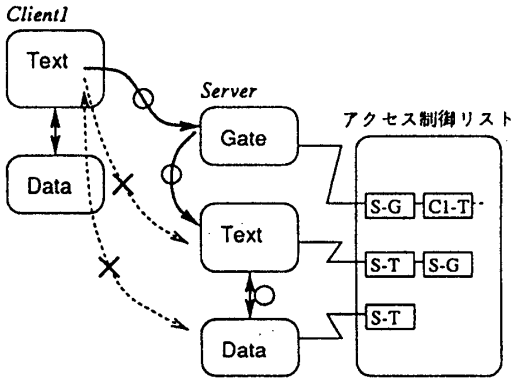

図 3: メモリ・セクションに関するアクセス制御

とアクセス属性の組を持たせる。

2. アクセス権チェック機構

   アクセス時には、アクセスを発生した主体、起源、属性をアクセス先ページのアクセス制御リストに照合して許可を得る。TLB内には、ページ毎に使用頻度の高いアクセス権を複数保持して、アクセス権チェックを高速化する。

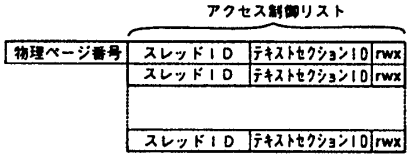

図 4: ページ・テーブル・エントリの構成

4 まとめ

本データ保護方式は、64bitアドレス空間の大きさを活かして、クライアント／サーバ型プログラミング・モデルを保ったまま、実行時のOSのオーバヘッドを削減し、高性能な分散環境を実現することができる。

参考文献

[1] John R. Mashey, "64-bit Computing," *Byte*, pp.135-142, September 1991.

[2] Brian N. Bershad, *et al.*, "User-Level Interprocess Communication for Shared Memory Multiprocessors," *ACM Transactions on Computer Systems*, pp.175-198, Vol.9, No.2, May 1991.

[3] "SunOS Programming Utilities & Libraries," Sun Microsystems, Inc., March 1990.

[4] Jeffrey C. Mogul and Anita Borg, "The Effect of Context Switches on Cache Performance," *ASPLOS-IV Proceedings*, pp.75-85, April 1991.

[5] 申承昊、他、"大規模アドレス空間を利用するOSの構想" 情報処理学会第44回全国大会予稿集, March 1992(予定).