

## 次世代アーキテクチャ向けオペレーティングシステム

1V-1

## マイクロカーネルの評価

津田 悦幸      福本 淳      寺本 圭一      友田 一郎      岡本 利夫

(株) 東芝 研究開発センター

## 1 はじめに

既報したとおり、我々は、次世代マイクロプロセッサアーキテクチャ向けオペレーティングシステム Cubix (CUBe of 2 byte unIX) の開発を行っている。[1, 2, 3, 4]

このほど、次世代マイクロプロセッサ上のオペレーティングシステムを開発する上での、プロトタイプを i386 上に試作し評価を行ったので、結果を報告する。

## 2 Cubix の主な特徴

- 単一仮想記憶 (SVS: Single Virtual Space)

Unix などでは、プロセスごとに仮想空間が独立な、多重仮想記憶であったが、Cubix では、次世代マイクロプロセッサの 64 ビットの広大な仮想空間上に、すべてのプログラムとデータを配置し、プログラム間で仮想空間を共有する (図 1)。

これによって、サーバー・クライアント間といったプロセス間通信は、データコピー、あるいは、データ領域のアドレス変換を行わずに、メモリポインタの受け渡しによって実行できるので、アプリケーション間の通信を高速に実行することができる。

- アクセス制御リスト (ACL: Access Control List) によるアクセス保護

図 1 のように、仮想空間は、複数のメモリセクションより構成され、386 におけるプロトタイプでは、アクセス制御リストのチェメモリセクション毎に、【スレッド ID】と【プログラムの存在するメモリセクション ID】をキーとして、read/write/execute の各属性を指定したアクセス制御リストを設定し、実行時、カーネルがアクセス制御リストをチェックすることによって、メモリセクションに対するアクセス保護を行う。

次世代マイクロプロセッサにおける Cubix では、ア

クセス制御リストのチェックは、MMU 等のハードウェアで行うことを仮定しているが、i386 におけるプロトタイプでは、アクセス制御リストのチェックはソフトウェアで行っている。

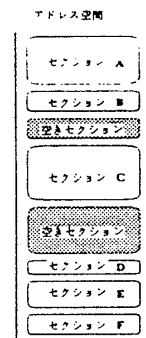


図 1: 仮想空間構成概念図

## 3 Cubix におけるプロセス間通信方法

Cubix では、上記したように、クライアント・サーバー間といった、アプリケーション間の通信を高速化するために単一仮想空間を用いているが、単一仮想空間のメリットを生かしたアプリケーション間のデータ通信方法として、以下の 2 通りのデータ通信方法が存在する。

- スレッド間通信

クライアント・スレッドとサーバー・スレッドが、スレッド間通信用のデータセクションを共有し、該共有データセクションに、通信データを書き込んだり読み出したりすることによって、スレッド間のデータ通信を行う (図 2)。

- スレッドのメモリセクション間ジャンプ

スレッドが、通信用のデータセクションを保持したまま、クライアントのテキストセクションから、サーバーのテキストセクションへジャンプすることによって (あ

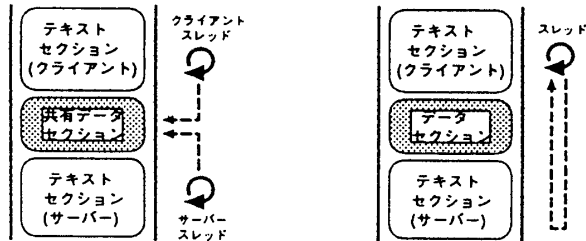
Evaluation of a Microkernel-based Operating System for Next-Generation Architecture

Yoshiyuki TSUDA, Atsushi FUKUMOTO, Keiichi TERAMOTO, Ichiro TOMODA, Toshio OKAMOTO

Toshiba Corporation, R&D Center

1, Komukai-Toshiba-cho, Saiwai-ku, Kawasaki 210, Japan

るいは、その逆)、アプリケーション間のデータ通信を行う(図3)。スレッドスイッチをする手間が省ける分、スレッド間通信よりも速いと考えられる。



<図2> Cubixにおけるスレッド間通信の概念図

<図3> Cubixにおけるスレッドのメモリセクション間ジャンプの概念図

#### 4 評価モデルと、評価結果

前記したように、Cubixでは、クライアント・サーバー間といった、プロセス間のデータ通信を高速で行えるように、すべてのプログラムとデータが、単一の仮想空間を共有している。仮想空間を共有化することによって、アプリケーション間のデータ通信が、どれ程高速化されるかについて、評価を行った。

本稿では、特に、クライアントの所有するデータをサーバーに転送し、サーバー側でデータを他の領域にコピーし、結果をクライアントに返すといった、Xクライアント・Xサーバーの動作を模擬した評価モデルを作成し、転送データサイズを変化させて、クライアントがデータを転送してからクライアントに結果が返って来るまでの時間を計測した。

また、Cubix上で、上記評価を実行した場合と、他のOS上で実行した場合を比較するために、Cubixで評価したのと同じマシン(486マシン)で、Unix上でSunRPCを用いて上記評価モデルを模擬した場合と、mach2.5上でIPCを用いて上記評価モデルを模擬した場合について、それぞれ計測し、比較を行った。結果を、図4に示す。

図4によれば、我々の開発したCubixのACLによるアクセス保護をハードで実現し、メモリセクション間ジャンプを用いた場合(cubix\_hard)、非常に高速にクライアント/サーバー・システムを構成できることがわかった。また、上記アクセス保護をソフトで実現した場合(cubix\_jump)も、mach(mach2.5\_ipc)と比較した場合、高速であることがわかった。

machとCubixのスレッド間通信(cubix\_comm)を比較した場合、Cubixは単一仮想記憶なので、アドレス変換をしない分、Cubixのスレッド間通信の方が高速であるはずであるが、現バージョンのCubixでは、スケジューラをカーネル外で実現しているため、Cubixは、それほど速くはな

い。参考として、Unix(unix\_rpc)とmachを比較した場合、データコピーの手間が幾分省略できる分、machの方が高速であった。

#### 5 おわりに

以上、図4で示したように、単一仮想記憶を採用することによって、アプリケーション間のデータ通信を高速に実現することがわかった。今後、さらに評価を進めるとともに、評価結果を基に、次世代マイクロプロセッサ上に実装するCubixの仕様を固める予定である。

#### 参考文献

- [1] Okamoto et al., "A Micro Kernel Architecture for Next Generation Processor", pp.83-94, Micro-kernels and Other Kernel Architectures Symposium, USENIX, 1992/4
- [2] 瀬川 英生 他, 「単一仮想記憶域を提供する64ビット・アドレス指向OS」, 情報処理学会 第4回コンピュータシンポジウム, 情処シンポジウム論文集 Vol.92, No.7, 1992/10
- [3] 福本 淳 他, 「次世代アーキテクチャ向けオペレーティングシステム・マイクロカーネルの開発」, 情報処理学会第47回大会, 6B-6, 1993/10
- [4] 瀬川 英生 他, 「単一仮想記憶域を特徴とする64ビット・アドレス指向OS」, 情報処理学会 第4回コンピュータシンポジウム, 情処シンポジウム論文集 Vol. 92, No. 7, 1992年10月.

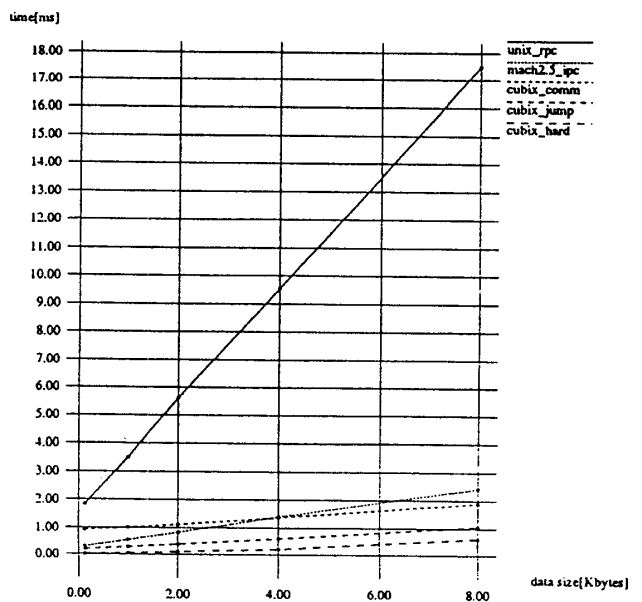


図4: アプリケーション間のデータ通信時間