汎用ホモジニアスマルチコアプロセッサにおける OS とスレッドライブラリ

佐藤　未来子[†]　磯部　泰徳[†]　十山　圭介[††]
野尻　徹[††]　入江　直彦[††]
内山　邦男[†††]　並木　美太郎[†]

1. はじめに

近年，ゲーム，カーナビゲーションシステム，携帯電話等の情報家電機器を始め，多くの計算機器にマルチコアプロセッサが搭載されている[1)~3)]．汎用コアを複数組み合わせたホモジニアス構成のマルチコアプロセッサに関しては，単純な汎用コアをチップ上に複数個搭載できることから多コア化，多機能化が進むと考えられる．今後進化するマルチコアプロセッサの機能を十分に活用して性能を引き出すためには，各ハードウェアアーキテクチャの特性に応じた最適なリソース管理やスケジューリング制御などが有効であると考える．本研究では，汎用ホモジニアスマルチコアプロセッサ (以下，マルチコアプロセッサ）を対象に，その性能を活かすためのプログラム開発環境やその実行基盤の実現を目指している．

2. 本研究の目標と方針

図 1 に，本研究で目標とするソフトウェア実行基盤の概念図を示す．本研究では，組込みや並列計算向けプログラムを対象として，マルチコアプロセッサの演算性能を引き出して高効率実行することを目標とする．

そのために本ソフトウェア実行基盤では，従来の汎用 OS でサポートされている POSIX 仕様のスレッド（PThread）を軽量に管理・制御する**並列演算向け専用 OS**(以下，専用 OS）を実現し，性能を追求する．また，本ソフトウェア実行基盤では並列演算向け専用 OS と従来の汎用 OS の両方を稼動させる**ヘテロジニアス OS 構成**とし，一つのアプリケーションプログラムを複数の OS の適材適所で処理することにより，実用性を追求する．例えば，専用 OS 上で実行中の並列プログラムでファイル入出力や GUI などの要求がある場合，それらの処理を汎用 OS 側へ委託できるようにする．これにより，専用 OS を肥大化・複雑化させずに I/O 関連の処理に対応する．

† 東京農工大学大学院
†† 日立製作所 組込みシステム基盤研究所
††† 日立製作所 研究開発本部

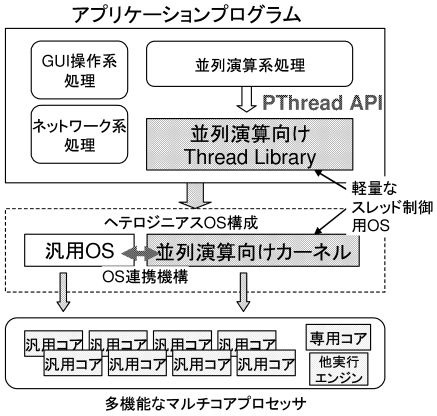

図 1　ヘテロジニアス OS 構成で実現するソフトウェア実行基盤
Fig. 1　The Software platform structured by heterogeneous OS

3. システム構成

図 2 に本研究のシステム構成を示す．マルチコアプロセッサの一コアを汎用 OS（本研究では SH-Linux）へ割り当て，残りのコアを専用 OS へ割り当てて並列プログラムを実行する．異種 OS 間で処理を連携させるための **OS 連携機構**[4)] を設け，並列プログラムの起動や，I/O 関連の処理を汎用 OS 側で担えるようにする．例えば，汎用 OS 側のファイルシステム上で管理している専用 OS 用の並列プログラムを，OS 連携機構を使って専用 OS 上で実行する．なお OS 連携機構の詳細に関しては別途報告する予定である．

専用 OS に関しては，筆者等がマルチスレッドアーキテクチャを対象に研究開発している OS「Future」と，マルチスレッドライブラリ「MULiTh（Userlevel Thread Library for Multithreaded architecture）」をマルチコアプロセッサ向けに一部改変して実装した．

3.1 MULiTh

MULiTh は POSIX スレッドを管理するスレッドライブラリであり，コアへのスレッド割当てやスレッド制御を担う．MULiTh で生成したスレッドは，同期やコア放棄等のタイミングで切り替わるが，コアの横取り制御（preemption）は行わない．

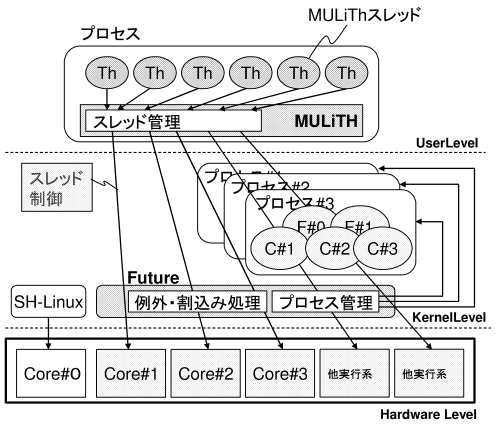

図 2 システム構成（4 汎用コアプロセッサの例）
Fig. 2 The System structure

表 1 情報家電用マルチコアプロセッサ仕様
Table 1 Spec of the multicore processor for consumer electronics

項目	仕様
CPU コア	SH-4A(600MHz) × 4 core
システムバス周波数	300MHz
命令キャッシュ	32KB(4way set-associative)
データキャッシュ	32KB(4way set-associative) (スヌープ機構有り)
命令用内蔵 RAM	8KB
データ用内蔵 RAM	16KB
命令・データ共有内蔵 RAM	128KB

3.2 Future

Future は MULiTh のスレッド制御をサポートするカーネルである．Future における「プロセス」とは，プロセッサ上の複数コアと，アドレス空間と，I/O を割り当てた実行単位である．MULiTh が管理するスレッドはプロセスに割り当てたこれらの資源を共有する．Future は MULiTh で行うコアへのスレッド割り当てには関与せず，プロセスへの実コア割り当てや，例外・割込み等の特権処理を担っている．

4. スレッドライブラリの基本性能

Future/MULiTh を NEDO の「リアルタイム情報家電用マルチコア技術の研究開発」プロジェクトで開発したマルチコアプロセッサ[3]（**表 1**）へ実装し，MULiTh の主な PThread I/F の基本性能を計測した．結果を**表 2** に示す．参考データとして，シングルコア向け SH-Linux（kernel-2.6.19）の PThread（Linux-Threads）の基本性能を同一プロセッサで計測した結果を示す．

MULiTh では，ライブラリ内部で直接コアの割り当てやスレッド制御を行うため，汎用 OS と比較してスレッド制御が軽量となっている．このことは，マルチコアプロセッサに対して本研究の専用 OS のアーキテクチャを適用することで，軽量なスレッド制御を実現できることを示している．また，汎用 OS でオーバヘッドが大きい `pthread_create/join` を多用するプログラムスタイルも採用できる．さらに，本研究のシステム構成であれば，MULiTh のスレッドスケジューラをプロセスごとに変えられる．並列プログラムごとに適したスケジューラを採用することで，並列プログラムの特性に応じた性能向上が期待できる．

表 2 基本性能の計測結果
Table 2 Results of the basic funciton on MULiTh

計測項目	MULiTh (μ sec)	Linux (μ sec)
`pthread_attr_init`	0.024	0.082
`pthread_attr_setdetachstate`	0.023	0.032
`pthread_create(1core)`	3.38	666.8
`pthread_create(4core)`	3.84	—-
`pthread_mutex_lock(1core)`	0.113	0.385
`pthread_mutex_lock(4core)`	0.238	—-

5. まとめと今後の課題

汎用ホモジニアスマルチコアプロセッサで並列プログラムの実行性能を引き出すためのソフトウェア実行基盤について述べた．今後も，多コア化，多機能化に備えて，Future のプロセス管理や MULiTh の機能強化を図る．

謝辞

本論文におけるマルチコアチップ及び実行環境は，NEDO リアルタイム情報家電用マルチコアプロジェクトにて早稲田大学，(株）ルネサステクノロジ，(株）日立製作所により開発されたものです．関係者の皆様に感謝申し上げます．

参 考 文 献

1) Pham, D. et al.: The Design and Implementation of a First-Generation CELL Processor, *2005 IEEE ISSCC*, Vol. 1, pp. 184–592 (2005).
2) Torii, S. et al.: A 600MIPS 120mW 70uA Leakage Triple-CPU Mobile Application Processor Chip, *the IEEE ISSCC Digest of Technical Papers*, pp. 136–137 (2005).
3) 早瀬清ほか: 独立に周波数制御可能な 4320MIPS, SMP/AMP 対応 4 プロセッサ LSI の開発, 情処研報, Vol. 2007, No. 55, pp. 31–35 (2007).
4) 磯部泰徳ほか: マルチコアプロセッサにおけるスレッドライブラリと OS の連携方式の設計, 情報処理学会第 70 回全国大会 3P-9 (2008).