マルチコアプロセッサにおける スクラッチパッドメモリを用いたメモリ管理システムの試作

北原悠† 佐藤未来子‡ 磯部泰徳‡ 並木美太郎§
東京農工大学工学部情報コミュニケーション工学科 †/東京農工大学大学院工学府情報工学専攻 ‡
東京農工大学大学院共生科学技術研究院 §

1 はじめに

近年、マイクロプロセッサの主流が、消費電力の問題と微細化の進展という 2 つの背景から、シングルコアからマルチコアへと変遷しつつある。それはパソコンやサーバーだけの話にとどまらず、組込み機器の分野においても同じ事が言える。

特に消費電力に関しては携帯型製品にとっては重要な問題となることが多く、消費電力の大きいキャッシュメモリの代わりに、低電力のスクラッチパッドメモリを用いるといった研究も以前からしばしば行われている。それらの研究によると、スクラッチパッドメモリ (以下 SPM) を用いることは消費電力のみならず、性能向上にも効果がある事がわかっている [1]。

しかしながら、既存の研究の多くはコンパイラでプログラムを最適化することにより SPM 上に静的にコード・データを割り当てる方式や、動的に SPM を割り当てるためのコードを事前に挿入し、実行時に DRAM から SPM へのメモリ割り当てをする方式がほとんどである [2]。そのため、SPM のサイズに限定されたメモリ割り当て方式しか行えず、SPM 管理のためのコードを挿入する特殊なコンパイラサポートが必要であった。また、基本的にこれらはマルチコアプロセッサでの適用は考慮されていない。

そこで本研究では、OS によって SPM を動的に管理する事によって、特定のコンパイラなどに依存することなく、様々なプログラムをマルチコアプロセッサで実行できる環境を提供するためのメモリ管理システムの試作と評価を行った。

2 本研究の目標

本研究では、SPM を OS により動的にページ管理することの有用性を検証し、MMU を用いて OS における動的な SPM の割り当て管理方式を試作する。また、マルチコア環境においては本来ローカル用途で用いられることがほとんどであった SPM を、他コアからもアクセス可能にするメモリ管理方式を適用し、その効果を検証する。

最終的には、本研究で提案するメモリ管理システムを用いることによってプログラム実行時の性能向上と消費電力増加の抑制を目標とする。

3 本システムの概要

本システムは、OS の一部としてメモリ管理を担うので、ユーザはメインメモリとスクラッチパッドメモリの存在を意識する必要がない。よって、特殊なコーディング手法や、特定のコンパイラなどによる最適化を行わずとも、どんなプログラムでも実行時にスクラッチパッドメモリへの割り当てが行われ、性能向上するように運用される。

本研究では、マルチコアの管理とマルチスレッドプログラミング環境を提供するだけの専用 OS 上にこのメモリ管理システムを組み込んだ。

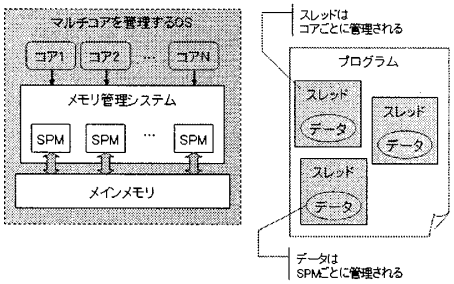

図 1: システムの概要

4 設計

4.1 階層式メモリ管理

OS によって動的に SPM 管理する手法として図 2 に示すような階層式メモリ管理手法を提案する。

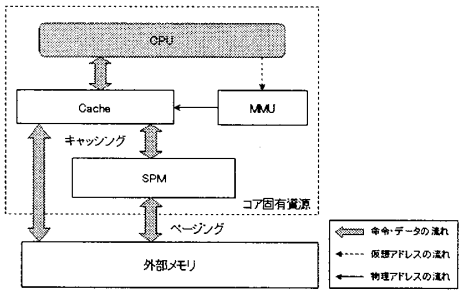

図 2: 階層式メモリ管理のイメージ

階層式メモリ管理では、SPM と外部メモリの間でページングを行う。CPU が扱うのは基本的に仮想アドレス空間であり、それが MMU によって物理アドレス空間に変換される。

命令・データはキャッシュを経由して高速にやり取りされるが、この際に一部の命令・データは SPM に転送され、ページングされる。

このようにキャッシュと外部メモリの間にもう一段、SPM を挟んでデータアクセスのレイテンシを低下させようとする試みが、階層式メモリ管理である。ただし、データのアクセスパターンやワーキングセットによっては、SPM への転送オーバーヘッドが無視できないレベルまで肥大化してしまう場合もある。

その場合、ページフォルトハンドリング中に性能劣化を動的に検出し、それ以降の処理では SPM へのデータ転送は行わずにキャッシュのみを用いて、直接外部

Prototyping of Memory Management System with Scratchpad Memory for Multi-core Processor

† Yu KITAHARA
Department of Computer, Information and Communication Sciences, Tokyo University of Agriculture and Technology
‡ Mikiko SATOH, Hironori ISOBE
Department of Computer and Information Sciences, The Graduate School at Tokyo University of Agriculture and Technology
§ Mitaro NAMIKI
Institute of Symbiotic Schience and Technology, The Graduate School at Tokyo University of Agriculture and Technology

メモリを参照するという従来どおりの運用も可能とする。転送が抑制されていても性能劣化の改善が見られたようならばその時点から SPM へのデータ転送を再開できるものとする。

4.2 SPM 空間

図 3 に示すように、すべてのコアのそれぞれの SPM を、あわせて 1 つの SPM 空間であるとみなして管理する。ただし、個々の SPM に対してページの割り当てを行えるのは、その SPM を保有しているコア自身のみとする。

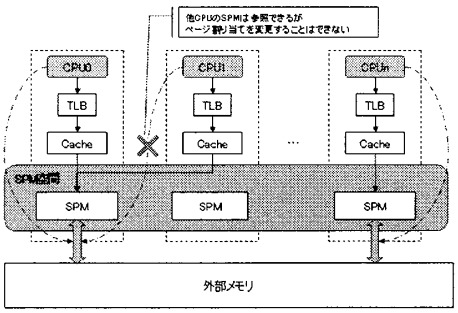

図 3: SPM 空間の管理モデル

ヒープ、グローバルデータなどはコア全体で共有する頻度も多く、個々のコアでアクセス要求が発行される度に割り当てを行うより、一度割り当てられたデータを再利用することで、データ転送を抑制する事につながり、パフォーマンスの向上が期待できる。特に、同一のデータを別々の SPM に割り当ててしまうと整合性をとるために多大なオーバーヘッドが生じる可能性があるので、このような一元管理の方がコストは少ないと考えられる。

5 実装と評価

本研究では、階層式メモリ管理方式の基本構造と、SPM 空間として SPM を扱える構造の大枠の実装を行った。その結果、メモリ管理の核となる部分のバイナリサイズは約 7KB になった。カーネル全体のサイズが約 154KB なのでカーネルの 4.5%を占めることとなった。

本メモリ管理システムをホモジニアスマルチコアアーキテクチャRP1 プロセッサ [3] 上に実装し、実行時性能と消費電力についての評価を行った。ベンチマークには n*n の正方行列同士の掛け算を行う matrix を用いた。実行サイクル数と消費エネルギーについて、キャッシュメモリ単体でプログラムを実行した場合と、本メモリ管理システムを用いてキャッシュメモリと SPM を協調動作させた場合の性能比率について、1 コア動作時を図 4 に、4 コア動作時を図 5 に示す。

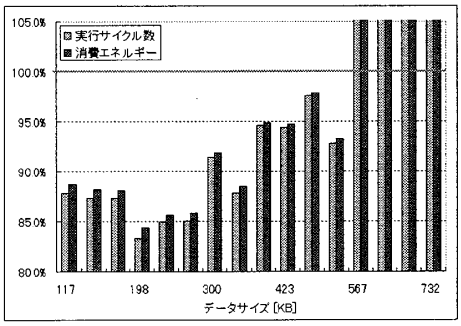

図 4: 従来管理方式と本システムとの性能比率
(1 コア動作時)

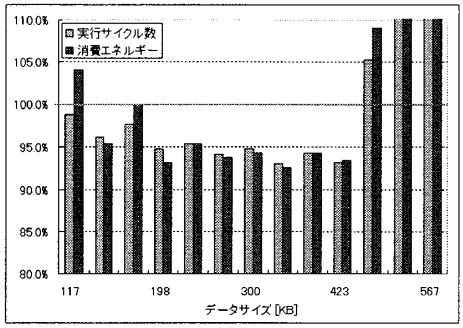

図 5: 従来管理方式と本システムとの性能比率
(4 コア動作時)

RP1 プロセッサの SPM は 1 コアあたり 128KB である。よって図 4 より、ユニコア環境で本メモリ管理システムを運用した場合、データサイズが SPM サイズの 4 倍程度までならキャッシュメモリ単体で運用したときよりも性能が向上しているといえる。それ以上のデータサイズで軒並み性能が低下しているのは、性能劣化の検出に問題があるためだと考えられる。また、図 5 より、マルチコア環境下では SPM 空間により見かけ上の SPM サイズは 4 倍になっているはずだが、1 コア運用時にみられたような性能向上率は得られなかった。これは、コアごとにデータの整合性保証を行う際の排他処理などのオーバーヘッドが密接に関係していると考えられる。

実行時性能について、ユニコア環境で平均 10%、マルチコア環境で平均 5%の改善がみられた。これらの結果はさまざまなベンチマークを用いたさらなる評価が必要であるが、今回得られたデータは OS で SPM を動的に管理することの有用性を示す糸口になっているといえる。

6 おわりに

本論文では、スクラッチパッドメモリの動的割り当てを用いたマルチコアプロセッサにおける階層式メモリ管理システムの設計と実装について述べた。

今後の課題は、複数ケースのベンチマークで評価を行い、さらなる性能改善のための最適化手法を研究することである。また、より多くのアプリケーションに柔軟に対応できるようにし、マルチプロセスへと対応も視野に入れている。

参考文献

[1] Bernhard Egger, Jaejin Lee, Heonshik Shinl. Scratchpad Memory Management for Portable Systems with a Memory Management Unit. EMSOFT '06: Proceedings of the 6th ACM & IEEE International conference on Embedded software, p321 - 330 (2006)

[2] Rajeshwari Banakar, Stefan Steinke, Bo-Sik Lee, M. Balakrishnan, Peter Marwedel. Scratchpad Memory : A Design Alternative for Cache On-chip memory in Embedded System. ACM CODES '02: Proceedings of the tenth international symposium on Hardware/software codesign, p73 - 78 (2002)

[3] 早瀬 清, 吉田 裕, 亀井 達也, 芝原 真一, 西井 修, 服部 俊洋, 長谷川 淳, 高田 雅士, 入江 直彦, 内山 邦男, 小高 俊彦, 高田 究, 木村 啓二, 笠原 博徳. 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, p31 - 35 (2007)