「ポストペタスケール高性能計算に資するシステムソフトウェア技術の創出」
平成22年度採択研究代表者

H24 年度
実績報告

堀 敦史

独立行政法人理化学研究所計算科学研究機構・上級研究員

メニーコア混在型並列計算機用基盤ソフトウェア

§1. 研究実施体制

(1)堀グループ(理化学研究所計算科学研究センター)

① 研究代表者:堀 敦史 (理化学研究所計算科学研究センター、上級研究員)

② 研究項目

・メニーコア用 OS カーネルの開発

・スケーラブル並列ファイルシステム

・超軽量マルチスレッド機構

(2)並木グループ(東京農工大)

①主たる共同研究者:並木 美太郎 (東京農工大学 大学院工学研究院、教授)

②研究項目

・メニーコア用 OS における資源管理と仮想化方式

(3)辻田グループ(近畿大)

① 主たる共同研究者:辻田 祐一 (近畿大学工学部、准教授)

② 研究項目

・高スケーラブルな通信と I/O の実現

(4)Dongarra グループ

① 主たる共同研究者:Jack Dongarra (University of Tennessee, Director)

② 研究項目

・故障レジリエンス

§2. 研究実施内容

本研究では、ポストペタスケールにおいて主流のひとつになると思われるメニーコアアーキテクチャをターゲットとし、各種 HPC アプリケーションのスケーラブルな動作が可能なシステムソフトウェアの研究開発をおこなうものである。このため研究範囲は広範であり、OS カーネル、メモリや通信の遅延を隠蔽するための軽量スレッド、低レベル通信機構と標準メッセージ通信ライブラリ MPI、ファイル IO などがその範疇となっている。昨年度は、設計の基礎となるべく各種基礎パラメータの計測、アイデアやモデルの検証、ソフトウェアの試作を目的としていたが、今年度は昨年度の成果を受けるだけでなく、入手可能になったメニーコア実機上での基礎評価、および、昨年度のアイデアのプロトタイプ実装と評価が中心となった。これらの成果を受け、来年度からは、実使用に耐えるソフトウェア群の研究開発につなげる予定である。

(1) メニーコア OS の通信機能と軽量スレッド（堀グループ）

(1-1) メニーコア OS におけるメモリ管理方式の検討

メニーコアでは文字通りコア数が一般の CPU より多い。また、今後予想される CPU アーキテクチャは現在のものよりより深いメモリ階層になると予想されている。本研究では、ソフトウェアによるメモリ階層の制御方式を検討した。その結果、メニーコアでは TLB 管理のためのオーバヘッドが非常に大きくなることが判明し、そのオーバヘッドを回避する方式を提案した。

(1-2) メニーコア用の新しいプロセスモデルの検討

メニーコアアーキテクチャは、基本的に多くのコアを並列動作させることでマルチコアの性能を上回ることができる。このためにはコア間通信の性能が重要になる。本研究では、従来の OS が提供するプロセスあるいはスレッドという概念では、メニーコアアーキテクチャにおいては様々なオーバヘッドが生じてしまうことを明らかにすると同時に、それらのオーバヘッドを解消するための新しいプロセス／スレッドモデルを提案した。

(1-3) メニーコアに向けた軽量スレッドに関する検討

Intel 社の Xeon Phi ではコアあたり 4 つのハイパースレッドをサポートしていると同時に、コアもハイパースレッドを使うことを前提に設計されている。本研究はメニーコアアーキテクチャにおけるハイパースレッドの特性を明らかにし、現在おこなわれているスレッド並列以外の方法で性能向上を目的とする軽量スレッドの研究である。今年度は Xoen Phi 上での基礎評価をおこない、その特性を明らかにした。来年度は、今年度得られたデータをベースにハイパースレッドの使い方について検討を進める予定である。

(2) メニーコア用 OS における資源管理と仮想化方式(並木グループ)

本グループでは、(a) 異種 OS 間を連携し、個々の資源を仮想化するプロセス管理、メモリ管理、通信管理などの資源管理の設計と実現 (b) マルチコア、メニーコアプロセッサに対して、スケーラビリティを確保し、応用プログラムに適応した資源割当てを行うスケジューラの研

究を行っている。本年度は、次のふたつの研究を行った。

(2-1) 異種 OS による実行基盤と資源管理の研究

異種 OS アーキテクチャにおける資源管理を実装し、異種 OS 構成法の基本原理を明らかにするための基礎を整備した。メニーコア側において計算を並列実行に特化させる基盤により高効率な環境を提供するとともに、メニーコア側では処理を行わない入出力をマルチコア側と連携する OS 間連携機構を Linux 上に実装した。本環境で、通信、ページングなどのメモリ管理、例外処理、入出力などについて、通信コスト、システム呼出しコストなどの評価を行った。

(2-2) メニーコア用軽量 OS の試作

メニーコアの並列性を活用するための軽量 OS として、ユーザレベルの仮想スレッドを提供する資源管理を試作した。本資源管理部を Intel 社メニーコアプロセッサ上で実行し、仮想化、同期、排他制御などのオーバヘッドを評価し、設計のための基本データを得ることができた。

今後は、並列性を高めるための資源割当てのアルゴリズムと方式、他グループの成果と連携しながら中間評価に向けた準備を行う。

(3) 高スケーラブルな通信と I/O の実現(辻田グループ)

メニーコアの各コアに出来る限り並列高性能計算に特化した処理を行わせるために、MPI ライブラリで管理する膨大な情報量の処理や集団型通信並びに集団型 I/O 操作をマルチコア側で処理する高スケーラブルな通信機構を検討してきた。この中で、マルチコア・メニーコア間で階層化された情報管理を行う実装やマルチコア側で集団型通信を行う delegation 機構のプロトタイプ作成と評価試験を行ってきた。メニーコア混在型並列計算機や、その上で動く OS 等が並行して研究開発中のため、PC クラスタ上に仮想的な混在型環境を用意し、この上で複数階層型コミュニケータ管理手法のプロトタイプ実装の開発と評価試験を実施した。その結果、delegation 機構がオリジナル実装よりも処理時間を短縮できるケースを確認できた。今後、実装の完成度を高めるべく、OS カーネル等の研究開発との連携を深めながら、より実用性を重視した通信機構の実現に向けて設計・実装を進める予定である。

(4) 故障レジリエンス機能(Dongarra グループ)

Over the first 6 months of the project we successfully moved our fault tolerance proposal from the stage of concept to the stage of implementation. The User Level Fault Mitigation (ULFM) presented in front of the MPI standardization body has been implemented in the context of the Open MPI implementation. This first version supports mostly Linux-based clusters, but the main concepts and implementation details should be portable to other environments, most particularly to the K computer.

For this first period the goal was to provide a version stable enough to become

usable by researchers outside our group, to spur investigations and innovations in the area of resilience. This goal has been successfully achieved; the current version, freely available on the Internet (http://fault-tolerance.org), has a growing community and a number of applications. We set up the website and corresponding software ecosystem to grow the community support and involvement.

The high-level goal for the next annual period is to port the entire framework (including both the runtime and MPI modifications) on the K computer and provide an optimized version of the main concept behind this proposal, the operations with a consensus behavior: the operation of revoke, agreement and shrink.

§3. 成果発表等

(3－1) 原著論文発表

●論文詳細情報

1. M. Sato, G. Fukazawa, K. Nagamine, R. Sakamoto, M. Namiki, K. Yoshinaga, Y. Tsujita, A. Hori, and Y. Ishikawa, "A Design of Hybrid Operating System for a Parallel Computer with Multi-Core and Many-Core Processors", In Proceedings of the 2nd International Workshop on Runtime and Operating Systems for Supercomputers (ROSS '12), Article 9, Venice, June 29, 2012.
2. Y. Ohno, A. Hori, Y. Ishikawa, "File Composition Technique to Improve the Performance of Accessing a Number of Small Files," Proceedings of the International Conference on Parallel, Distributed Processing Techniques, Applications, In , volume I, 2012.
3. A. Hori, T. Kameyama, Y. Tsujita, M. Namiki and Y. Ishikawa, “An Efficient Kernel-Level Blocking MPI Implementation,” Recent Advances in the Message Passing Interface, Lecture Notes in Computer Science, Vol. 7490, Springer, Vienna, September 2012, pp. 153-162.
4. K. Yoshinaga, Y. Tsujita, A. Hori, M. Sato, M. Namiki and Y. Ishikawa, “Delegation-Based MPI Communications for a Hybrid Parallel Computer with Many-Core Architecture,” Recent Advances in the Message Passing Interface, Lecture Notes in Computer Science, Vol. 7490, Springer, Vienna, September 2012, pp. 47-56.
5. B. Gerofi, A. Shimada, A. Hori and Y. Ishikawa, “Partially Separated Page Tables for Efficient Operating System Assisted Hierarchical Memory Management on Heterogeneous Architectures,” CCGrid 2013, Delft (to appear).