XeonPhi 搭載計算機における Multiple PVAS を応用した MapReduce フレームワーク

佐藤 未来子 †　　並木 美太郎 †

1. はじめに

近年，ビッグデータの処理基盤として MapReduce の並列分散処理フレームワークが注目されており[1]，GPGPU や Intel XeonPhi などのメニーコア向けフレームワークも提案されている[2][3][4]．本研究では，XeonPhi を搭載するヘテロジニアスアーキテクチャシステムを対象とした MapReduce フレームワークの処理性能向上を目指し，Multiple PVAS を応用した MapReduce フレームワークについて述べる．

2. 既存の MapReduce フレームワークと課題

既存の XeonPhi 向け MapReduce フレームワーク[3] [4]は，汎用 OS やランタイムライブラリで提供される並列実行基盤を利用して構築されている(図 1)．MapReduce の処理は，pthread ベースの master/worker モデルにより並列化され，両プロセッサへ配置する MapReduce に係るプロセス間では，MPI 通信を用いて同期やデータ転送を制御している．しかし，pthread では同一プロセッサ上での並列実行に限定されていること，また，既存のプロセッサ間通信のオーバヘッド等が及ぼす MapReduce フレームワーク処理性能への影響が懸念される．

3. 目標

XeonPhi を搭載するヘテロジニアスアーキテクチャシステムで性能向上を図るには，各プロセッサの適材適所へタスクを割り当て，プロセッサ間の同期制御やデータ転送制御を効率よく行うことが重要である．本研究では，既存の並列実行基盤を用いずに，MapReduce に係るタスクが互いのデータを直接アクセスするためのシステムソフトウェアにより，MapReduce フレームワークのタスク間の同期制御やデータ転送制御の効率化を図り，MapReduce 性能を向上させることを目標とする．

† 東京農工大学大学院

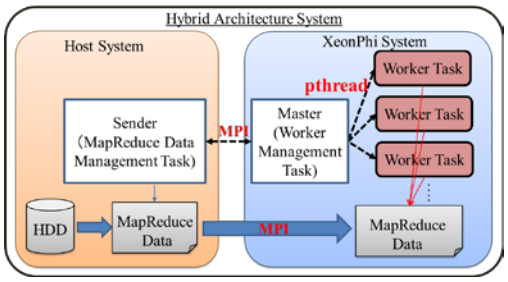

図 1　XeonPhi 向けの MapReduce フレームワーク

4. M-PVAS による MapReduce フレームワーク

4.1 設計方針

本研究では，これまでに XeonPhi を搭載するヘテロジニアスアーキテクチャシステムを対象としたプログラム実行基盤 Multiple PVAS（M-PVAS）を提案している[5]．M-PVAS では，異なるプロセッサ上のタスク同士が少ないオーバヘッドで連携するために，単一のグローバル仮想アドレス空間を提供し，異なるプロセッサ上のタスク同士であっても仮想アドレスを用いて互いのデータを自由にアクセスできるプログラム実行基盤である(図 2)．本研究では，M-PVAS 上で MapReduce に係るタ

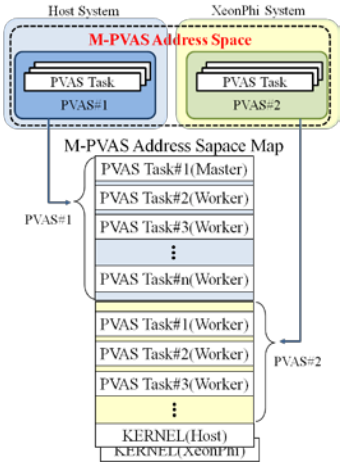

図 2　M-PVAS のタスクモデル

スクを，ホスト・XeonPhi 上の単一アドレス空間で並列実行させることにより，既存の MapReduce フレームワークで利用していた並列実行基盤を用いずに，目標とする MapReduce フレームワークを実現する．M-PVAS のタスク間の同期制御やデータ転送制御をメモリ参照ベースで行うことが可能となり，これが制御オーバヘッドを軽減させ，MapReduce 処理性能の向上につなげられる．

4.2 Master/Worker タスク制御の設計

M-PVAS による MapReduce フレームワークでは，アプリケーションの特性に応じて MapReduce を実行させるプロセッサと並列度をプログラマが自由に指定できる設計としている．本フレームワークでは，master/worker を PVAS タスクで実現し，両プロセッサのどちらで実行させたとしても，master/worker 間，および，worker 同士が互いのアドレス空間を参照しながらのタスク制御が可能なため，任意のタスク配置が可能である(図３)．

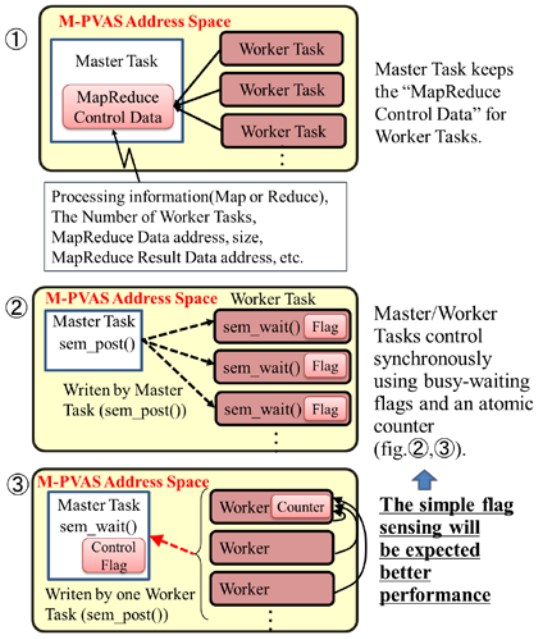

図３　M-PVAS における master/worker 制御

4.3 MapReduce 処理対象データ転送機能の設計

XeonPhi 上では扱えないファイルベースの大容量データも，XeonPhi 上の Worker において処理できるようにするために，ホストで管理する大容量データを，MapReduce 処理とオーバラップさせながらホストから XeonPhi へメモリコピーする機能を本フレームワークで提供する．

5. 実装と評価

設計に基づく M-PVAS を応用したフレームワークを，既存の pthread ベースのフレームワーク[4]約 13000 行のコードに対して約 1 割のコード改変を施し，図 1 に示した MapReduce フレームワークと同様モデルで実装した．文献[4]に付属のベンチマークの中から XeonPhi での並列実行により性能向上の効果がある「MonteCarlo シミュレーション」を用いて MapReduce 処理性能を比較した．

・pthread スレッドから M-PVAS タスクへの変更
・MapReduce 処理とデータ転送のオーバラップ

の二点の要因により MonteCarlo ベンチマークの MapReduce 処理において 1.8～2.0 倍の性能向上が図れた．

6. おわりに

本研究では，XeonPhi を搭載するヘテロジニアスアーキテクチャシステム向けプログラム実行基盤である Multiple PVAS を応用した MapReduce フレームワークについて述べ，MPI 通信や pthread 並列化を施した既存の MapReduce フレームワークとの性能比較について示した．今後も評価対象の MapReduce アプリケーションを増やして評価を続ける．

謝辞　本研究は，JST CREST における研究領域「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」研究課題「メニーコア混在型並列計算機用基盤ソフトウェア」によるものである．

参　考　文　献

[1] Welcome to Apache Hadoop (online), available from http://hadoop.apache.org.

[2] B. He, et al., "Mars: a mapreduce framework on graphics processors", in PACT, pp. 1-8, 2008.

[3] J. Talbot, et al., "Phoenix++: Modular MapReduce for Shared-Memory Systems", In the Second International Workshop on MapReduce and its Applications (MAPREDUCE), San Jose, CA, June 2011.

[4] M. Lu, et al., "Optimizing the MapReduce framework on Intel Xeon Phi coprocessor," Big Data, IEEE International Conference on, pp.125-130, Oct. 2013.

[5] M. Sato, et al., "Design of Multiple PVAS on InfiniBand Cluster System Consisting of Many-core and Multi-core", in EuroMPI/ASIA'14, pp.133-138, 2014 .