2K-9

# マルチコア・メニーコア混在型計算機による 高性能計算向けアプリケーション実行基盤「Multiple PVAS」の性能評価

深沢 豪† 佐藤 未来子† 吉永 一美‡ 辻田 祐一‡ 島田 明男‡ 堀 敦史‡
並木 美太郎†

†東京農工大学工学府情報工学専攻 ‡理化学研究所計算科学研究機構

## 1 はじめに

スーパーコンピュータの性能は日々向上し続けており，近年ではシステムに搭載するコア数を増加させて高性能化を図っている．筆者らは同一ノード上にマルチコア CPU とメニーコア CPU を混在させた「マルチコア・メニーコア混在型計算機」において，高性能計算の性能向上を目的としたシステムソフトウェアの研究開発を進めている．対象とする計算機では Intel 社の Xeon Phi のようにマルチコアとメニーコアが PCI Express バスにより相互接続され，各 CPU が備えるメモリに対する相互アクセスが可能である．本計算機ではマルチコアとメニーコアが異なる演算性能の特性を持つことから，CPU の特性に応じたタスク並列化と軽量なコア間通信が可能なプログラム実行基盤が重要となる．

本研究ではマルチコアとメニーコアを併用する新しいスタイルで実アプリーションを構築し，高い演算性能を実現するためのプログラム実行基盤 Multiple PVAS[1] を提案している．本論文では，Multiple PVAS の概要と Multiple PVAS を用いた Infiniband デリゲーションによるノード間通信性能の評価結果について示す．

## 2 PVAS と Multiple PVAS

PVAS（Partitioned Virtual Address Space）はプロセスとスレッドの中間に位置する新しいタスクモデルである．PVAS においてコアを割り当てる実行実体であるプロセスを “PVAS Task” と呼び，図 1 中央に示す “PVAS 空間” という一つの仮想アドレス空間に PVAS Task の各アドレス空間を配置し，ページテーブルを共有する．これにより，PVAS Task 間では共有メモリを確保することなく，仮想アドレス参照による Task 間データ共有を可能としている．

Multiple PVAS はマルチコア，メニーコアごとに管理される PVAS 空間を包括する形へ拡張したタスクモデルである．図 1 左端の Virtual Address Map に示すとおり，異なる CPU の PVAS 空間を “Multiple PVAS 空間” という単一の仮想アドレス空間へマップし，Multiple PVAS 空間に属する PVAS Task をすべて同じ仮想アドレス空間において実行可能とする．これにより，Multiple PVAS 空間内の仮想アドレスによる情報の受け渡しが，異なる CPU 上の PVAS Task 間でも行えるようになり，複数 CPU で稼働する PVAS Task を自由に構成できる．

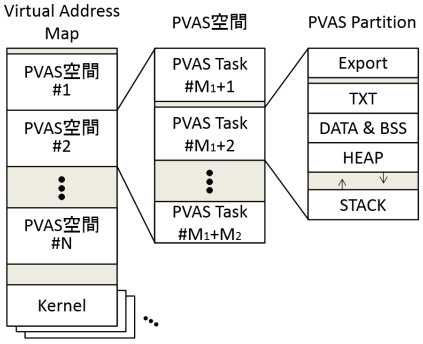


図 1: Multiple PVAS の仮想アドレス空間 [1]

Multiple PVAS ではリモート CPU の物理メモリへのアクセスに低遅延な MMIO を用いることで，CPU 間での処理依頼オーバヘッドを削減している．PVAS 空間ごとに構築するページテーブル間の一貫性を保つことで大域アドレス空間を実現するため，ページ単位かつ CPU をまたいだメモリアクセスが行われたエントリのみ一貫性を維持するほか，リモート CPU の物理メモリ上のページテーブルを直接参照するアドレス変換方式により一貫性維持のオーバヘッドを最小限にとどめている．

## 3 ノード間通信のデリゲーション

高効率に HPC アプリケーションを実行するためには高いノード間通信（I/O）性能が必要となる．マルチコア・メニーコア混在型計算機において，メニーコアから I/O 機器を直接操作する方式が提案 [2] されているが，I/O のソフトウェアスタックをコア単体性能の低いメニーコアで実行するため，高いコア単体性能を有するマルチコアと同等のノード間通信性能を得ることができない．Multiple PVAS による大域仮想アドレス空間を用いて，メニーコア上のアプリケーションが要求した I/O をマルチコア上のタスクに代行処理させることで，メニーコア単独では実現できない高い I/O 性能を得られると考えている．高効率な I/O デリゲーションをマルチコア上で実現するためには CPU 間でのデータ転送遅延の削減が重要である．本研究ではノード間通信

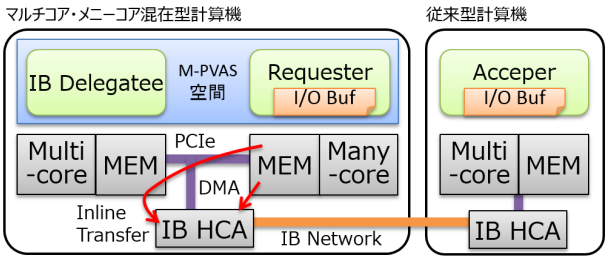


図 2: 評価システムの構成

に用いる IB HCA を対象とした I/O デリゲーションにおいて，メニーコアのメモリと IB HCA 間の直接データ転送（DMA）を Multiple PVAS でサポートすることで，HPC アプリケーションで広く利用されている MPI および IB のデータ転送遅延を削減する．

## 4 IB デリゲーションの性能評価

Multiple PVAS によるノード間通信のデリゲーションを実現する上での指標を得るため，Infiniband（IB）によるノード間通信をメニーコアからマルチコアへデリゲーションする場合の通信性能を測定した．MPI と IB を用いた多階層のノード間通信ではデリゲーションする処理の内容に任意性がある．本評価では最下層レイヤーとなる IB をデリゲーションする場合の通信性能を測定し，上位レイヤーのデリゲーションの必要性を検討する．

評価システムの構成を図 2 に示す．Multiple PVAS は Intel 社のメニーコア CPU である Xeon Phi を対象に実装され，IB への I/O を要求するメニーコア上の Requester と，IB への I/O を代行処理するマルチコア上の Delegatee が同一 Multiple PVAS 空間上で動作する．IB で接続された従来型計算機上では通信相手となる Accepter が動作する．MPI ライブラリの実装に使われる “send” と “write” の 2 種類をデリゲーションする場合の所要時間を測定し，メニーコア単独で処理する場合（Intel 社の CCL を利用）との性能差を比較する．メニーコアのメモリから IB HCA 間では DMA による直接データ転送が可能であるが，小サイズの転送では DMA を用いない “インライン転送” を実施することで通信遅延を削減する．インライン転送ではマルチコアのメモリを介したデータ転送を行なう．

図 3 に評価結果を示す．破線で示す “Only IB” はデリゲーションによる I/O において，マルチコア上での処理に要した時間を示している．デリゲーションによる I/O の所要時間は，転送サイズ 4KB 未満で CCL よりも大きく，4KB 以上ではほぼ同一となった．破線で示すマルチコア上での処理時間は CCL よりも平均 1us 程度削減されたが，デリゲーションに必要となる CPU 間通信に平均 1.7us 程度を要しているため，最小 0.4us,

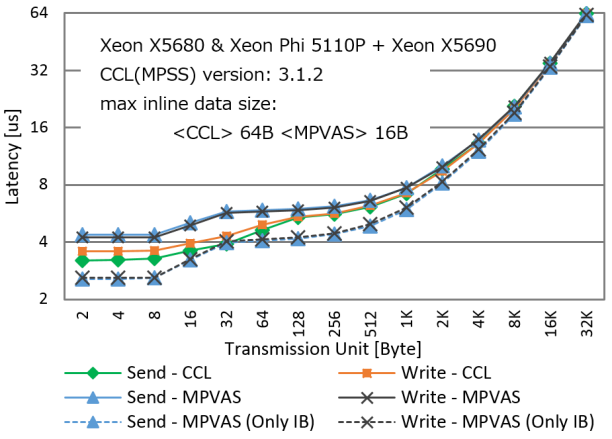


図 3: IB 通信性能の評価結果

最大 1.9us 程度 CCL よりも所要時間が増大する結果となった．

本評価結果より，Multiple PVAS の大域仮想アドレス空間を用いて高効率なノード間通信を実現するためには，IB よりも粒度の大きい I/O 処理をメニーコアからマルチコアへデリゲーションする必要があるといえる．たとえば MPI の集団通信をデリゲーションすれば，1 回のデリゲーションで複数回の IB 通信が行われるため，マルチコアでの高速な I/O 処理により削減される所要時間が，CPU 間通信のオーバヘッドを上回り，メニーコア単独での I/O 処理を上回るノード間通信性能を実現できると考えている．

## 5 おわりに

本論文ではマルチコア・メニーコア混在型計算機向けのタスクモデルである Multiple PVAS を用いた，IB デリゲーションの性能評価結果を示した．今後は IB よりも処理粒度が大きい MPI 通信をデリゲーションすることで，ノード間通信性能およびアプリケーション実行性能向上を目指す．

**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] 深沢豪，他: メニーコア混在型並列計算機向け大域仮想アドレス空間モデル Multiple PVAS の提案，情処 HPC 研究報告，Vol. 2013-HPC-141, No. 7, pp. 1–10 (2013).

[2] S. Potluri, et al.: MVAPICH2-MIC: A High Performance MPI Library for Xeon Phi Clusters with InniBand, Extreme Scaling Workshop 2013 (online) (2013).