

VI. 高性能計算システム研究部門

1. メンバー

教授	朴 泰祐、佐藤 三久、児玉 祐悦、高橋 大介
准教授	建部 修見
講師	川島 英之
助教	多田野 寛人
研究員	田中 昌広、梅田 宏明、松本 和也、Mihamed Amri Jabri
学生	大学院生 21 名、学類生 8 名

2. 概要

本研究部門では、高性能計算システムアーキテクチャ、並列プログラミング環境、GPU利用技術、並列数値処理の高速化研究、広域分散環境におけるデータ共有を中心とするグリッド計算技術等の研究を行っている。

3. 研究成果

【1】演算加速機構・通信機構融合に基づく並列演算加速機構に関する研究（朴、児玉）

JST-CREST 研究「ポストペタスケール時代に向けた演算加速機構・通信機構統合環境の研究開発」において、密結合演算加速装置（Tightly Coupled Accelerators: TCA）提案に基づく PEACH2（PCI Express Adaptive Communication Hub ver.2）FPGA チップを高度化し、そのシステムソフトウェアを充実させると共に、前年度までの point-to-point 通信の基本性能向上に加え、並列アプリケーションで必要となる代表的な collective 通信の実装と評価、さらにこれに基づく典型的な並列 GPU アプリケーションの性能評価を行った。また、PEACH2 の各種応用として、TCA ネットワークと従来の InfiniBand の併用による大規模並列向け通信システムの実装、GPU からの直接ノード間通信ライブラリの開発、Intel Xeon Phi への適用、Verbs API の PEACH2 への実装をそれぞれ行った。

図 6-1 に PEACH2 ボードの写真を示す。ボード中央に放熱フィンと共に FPGA（Altera Stratix-IV）が設置され、DRAM メモリ、他のノードとの外部 PCIe リンク接続のためのポート（計 3 ポート）とマザーボードに接続される PCIe スロットを備えている。ボードの右側 1/3 程度のエリアは安定化電源のためのレギュレータ類である。文部科学省特別経費による超並列 GPU クラスタ HA-PACS/TCA の全 64 台のノードにこの PEACH2 ボードを実装済みであり、このシステムを用いて本研究を継続している。HA-PACS/TCA のノードは、PEACH2 の PCIe 仕様に基づくハードウェア制約により、16 ノードずつ計 4 つのサブシステムに分割されている。

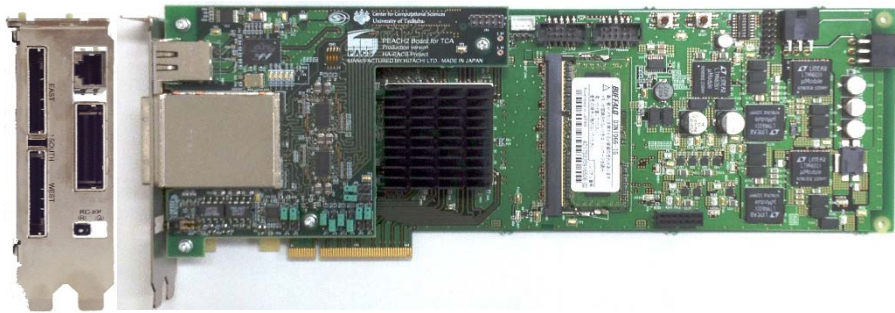


図 6-1 PEACH2 ボード (量産型)

(1) Collective 通信及び CG 法の性能評価

HA-PACS/TCA の 1 サブシステム (16 ノード) を用い、各種アルゴリズムと通信方式の組み合わせによって、Allreduce、Allgather 等の典型的 collective 通信の性能最適化を行った。

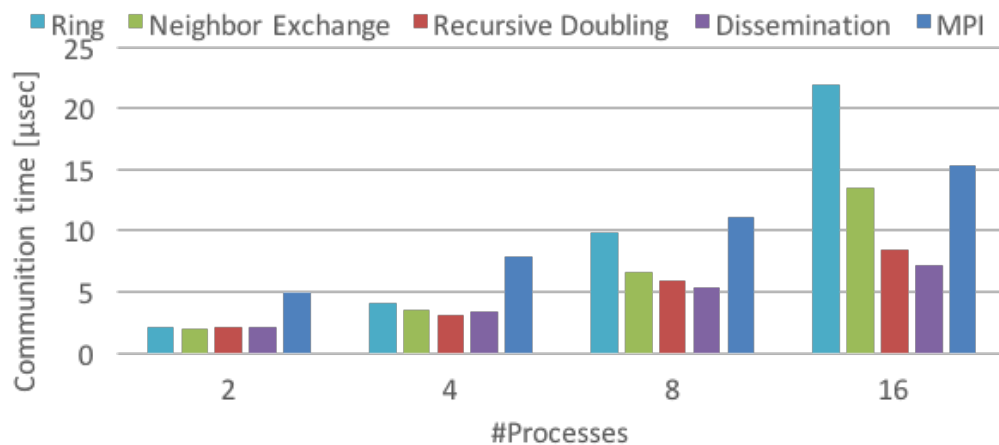


図 6-2 PEACH2 における Allreduce 性能と MPI/InfiniBand との比較

図 6-2 に、並列科学技術計算において多用される、倍精度実数 (8Byte) データの Allreduce の実行時間について、PEACH2 で各種アルゴリズムを用いた場合と、MPI/InfiniBand における実測値との比較を示す。16 ノードの場合、PEACH2 ネットワークで最適化された Dissemination アルゴリズムを用いた場合、通信レイテンシは MPI/InfiniBand の 50%以下に抑えられ、特に strong scaling において重要となる通信レイテンシの大幅短縮が行われている。

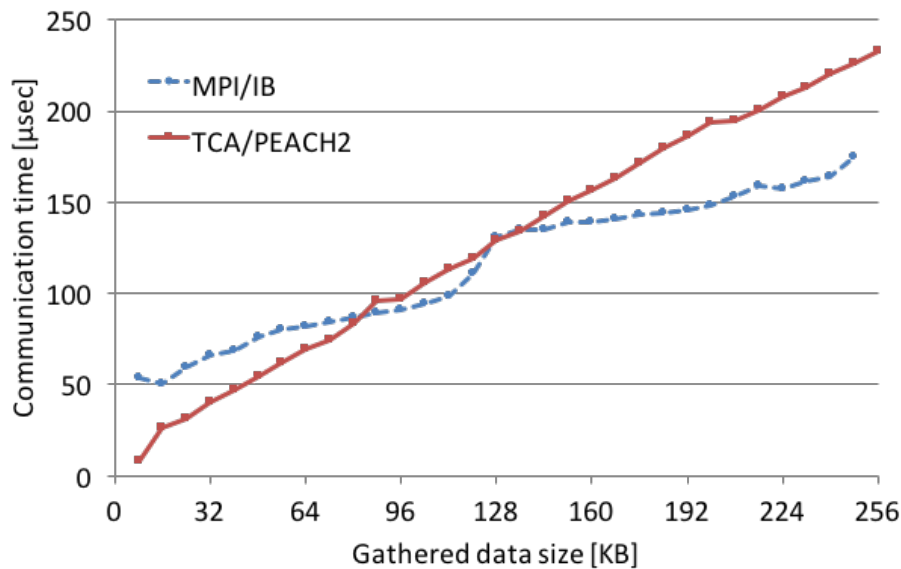


図 6-3 PEACH2 における Allgather 性能と MPI/InfiniBand との比較

図 6-3 に、16 ノードの場合の Allgather 通信の実行時間における PEACH2 と MPI/InfiniBand の比較を示す。ここではメッセージ長が 96KB までと短い場合は PEACH2 が約半分程度のレイテンシで通信できているが、メッセージ長が長くなると性能が逆転する。これは、PEACH2 のネットワークが 2 次元トラスであるのに対し、HA-PACS/TCA における InfiniBand が Fat-Tree ネットワークで無衝突通信可能であることが要因であるが、短メッセージ通信が重要となる strong scaling では PEACH2 が優位であることが示されている。

これらの基本 collective 通信ライブラリに基づき、一般疎行列に対する CG 法を HA-PACS/TCA 上で評価した。最大ノード数が 16 と比較的小さいため、単純な行方向 1 次元分割を行い、標準的な CG 法アルゴリズムを実装した。各ノード内での部分疎行列とベクトルの積 (SpMV) には CUSUPARSE ライブラリで提供されるルーチンをそのまま用いた。結果を図 6-4 に示す。これは中間サイズの行列に対する strong scaling の評価であるが、4~8 ノードにおいて PEACH2 の Allreduce 時間の短さが MPI 性能を上回ることによって貢献していることがわかる。しかし、Allgather ではノード数によって性能が逆転し、全体的にこの 2 つの通信の重さによって性能差に影響が出ることがわかる。また、大規模行列の場合、Allgather 通信のデータ長が大きくなるため、PEACH2 と InfiniBand でほぼ性能が互角になることも確認された。これらより、PEACH2 での strong scaling には適切な問題サイズと、Allgather アルゴリズムのさらなる改良が必要であることがわかった。

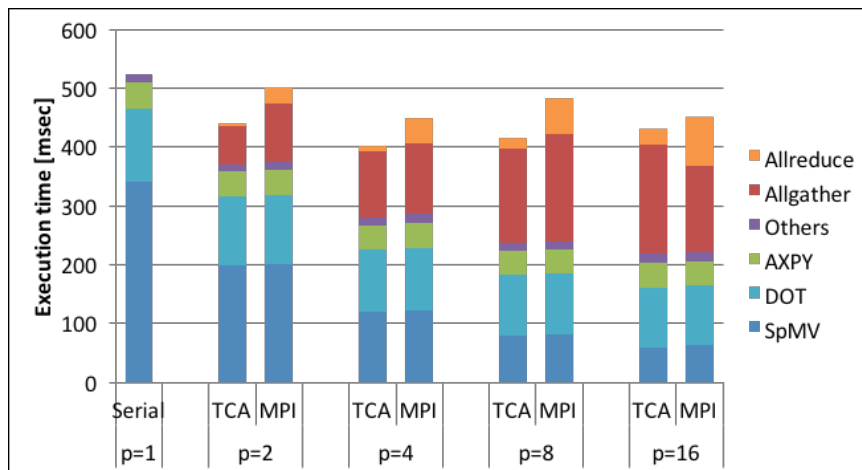


図 6-4 CG 法の 1 反復時間における PEACH2 と MPI/InfiniBand の性能差

(2) PEACH2 と InfiniBand の併用による大規模システム対応

PEACH2 の問題点の一つは、PCI Express による全てのノード、PEACH2、GPU をフラットに接続していることによるシステム規模の限界と、2 次元トラスによる通信距離の制限がある。より大規模なシステムに PEACH2 を適用するためには、従来の InfiniBand を併用した階層型クラスタの構築が必要となる。このため、中継点となるノードでは両ネットワークを併用し、通信の種類や方向に応じてこれらを使いわけ PEACH2/InfiniBand ハイブリッド通信システムを実装した。これを TCA 対応された XcalableACC コンパイラに実装することにより、ユーザ透過な形でハイブリッド通信システムを利用可能にした。

アプリケーション例として、HIMENO ベンチマークを 2 次元分割して 4 ノードにマッピングした例を示す。PEACH2 は block stride 通信に対応しているため、InfiniBand が苦手とする非連続領域の通信を PEACH2 に、連続領域通信を InfiniBand に、それぞれ機能分散的に割り当てることにより、通信の最適化が図れる。図 6-5 にその評価結果を示す。最大 16 ノードまでの分割方法と、MPI/InfiniBand のみ、PEACH2 のみ、ハイブリッドのそれぞれの通信時間を示している。多くの場合、InfiniBand と PEACH2 の単独通信の方が高速であるが、PEACH2 を多用できるいくつかの場合（1x8x1、2x8x1 等）では、ハイブリッド通信によって性能が向上している。これは、PEACH2 が unreachable であるノードと両ネットワークが使えるノードがこのマッピング上で混在した場合、混在ノード上での性能による全体の通信時間を短縮できる見込みがあることを示している。今後、さらにアルゴリズムを改良し、階層型クラスタで実際の性能を評価する予定である。

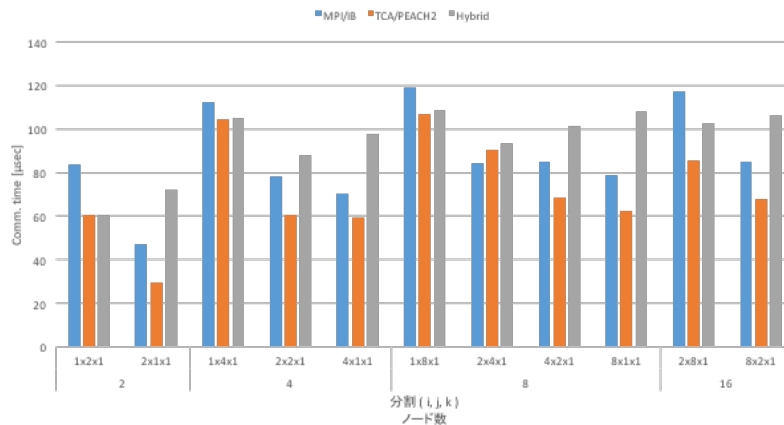


図 6-5 PEACH2 と InfiniBand のハイブリッド通信の性能

(3) GPU 間直接通信ライブラリの開発と PEACH2 への Verbs API の適用

CUDA 等の GPU プログラミングにおいて並列処理を行う場合、ユーザは基本的に通信と計算を切り離し、ホスト CPU からの offloading で GPU カーネルを起動、通信の度にホスト CPU に制御を戻さなければならない。このことは、MPI 化されている既存コードを CUDA に移植する場合の手間を非常に大きくしている。そこで、CUDA 環境で各 GPU 内から直接 MPI 通信を実行する通信システムとして GMPI (GPU direct MPI)を開発した。現実的には GPU から直接 PEACH2 や InfiniBand の通信トランザクションを起動することは困難であるため、ホスト CPU へのリクエストと応答処理でこれを実現し、ホスト CPU 上では通信を司るデーモンスレッドを走らせる。システムが正しく動くことを検証したが、ホスト CPU へのバッファリングと通信イベントのポーリングのオーバーヘッドが大きく、従来の MPI よりも性能が大きく落ちることが確認された。今後、通信方式を改良し、PEACH2 に適用することにより、性能を低下させることなく既存 MPI プログラムをスムーズに CUDA 化するシステムを開発していく。

また、InfiniBand 等で標準的に用いられる Verbs インタフェースを PEACH2 に実装した。これにより、PEACH2 の RDMA 機能の利用のために、ユーザプログラムの通信部分の全面的書き換えが必要な点を改良することと、PEACH2 をより広いシステム及びアプリケーションに適用することが可能になる。実装の結果、MVAPICH2 のような標準的 MPI が Verbs/PEACH2 上できちんと動作し、アプリケーションを実行可能であることが示された。しかし、現在のプロトタイプ実装では、PEACH2 が基本的に DMA write protocol のみを持つため、データ受信時におけるプロキシ通信のオーバーヘッドが大きく、InfiniBand に比べ通信レイテンシが大きくなることがわかった。今後、put/get のような PEACH2 との親和性の高い通信パターンでの性能評価を行い、さらに改良を加えていく。

(4) Intel Xeon Phi への PEACH2 の適用

GPU と並び、Intel Xeon Phi が新しい演算加速装置として注目されている。しかし、現在の Intel の開発環境では、Xeon Phi 間の通信に大きなボトルネックが存在することがわかっている。そこで、InfiniBand の代わりに PEACH2 を用いることにより、strong scaling における並列性能を向上させることが見込まれる。ただし、現在の Xeon Phi 周辺環境である MPSS では、InfiniBand 等の PCI Express デバイスを標準の x86 Linux と同様に見せることが難しい。そこで、MPSS で提供される SCIF インタフェースを用い、ホスト CPU にプロキシ動作をさせることで、PEACH2 を間接的にアクセスするシステムのプロトタイプを Xeon Phi 上に実装した。図 6-6 に SCIF 経由で PEACH2 を起動して通信した場合と、Intel MPI を用いた場合の性能比較を示す。これより、32KB 程度までの短メッセージ長において、Intel MPI を Xeon Phi 上に適用した場合に比べ、PEACH2 による通信が非常に低レイテンシで実現されることがわかる。

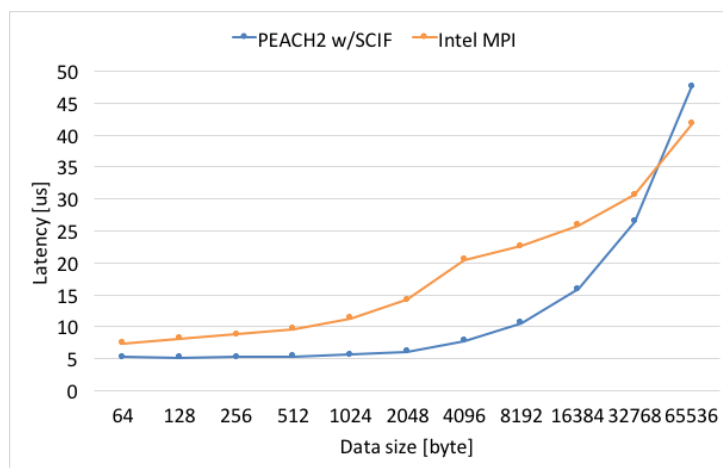


図 6-6 Intel Xeon Phi における SCIF 経由での PEACH2 通信の性能

【2】並列プログラミング言語 XcalableMP による核融合シミュレーションの高度化（朴、佐藤）

プリンストン・プラズマ物理研究所との共同研究に基づき、核融合シミュレーションコード GTC-P を並列記述言語 XcalableMP (XMP) のハイブリッド並列実装例としてコーディングを行った。XMP におけるハイブリッド並列とは、大域配列を全ノードで分割共有する global view と呼ばれるデータ分割と、各ノードのローカル配列を coarray 機能で直接参照する local view と呼ばれるデータ分割を併用する手法である。GTC-P は PIC (Particle In Cell) コードであるので、静的な空間（場）の物理の記述と、電子等の粒子が動的に動き回るという二種

類の振る舞いを記述する必要がある。そこで、前者に global view model を、後者に local view model をそれぞれ適用することにより、ユーザの物理的直感に即した並列プログラミングを実現することが可能となる。

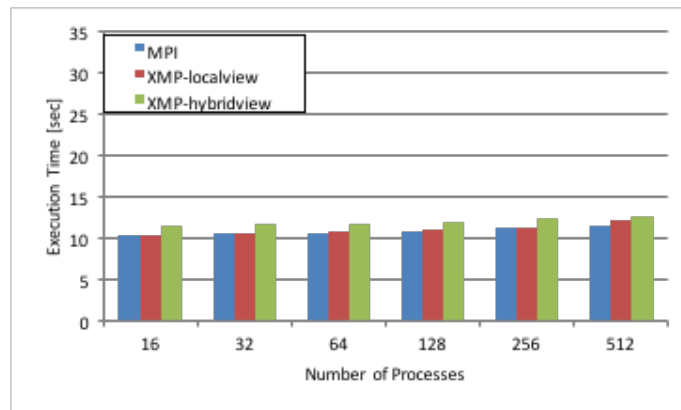


図 6-7 XMP における GTC-P のハイブリッド並列実装の性能評価

図 6-7 にトロイダル方向での並列分割を行った場合の、GTC-P の MPI による native 実装、XMP で local view を MPI 通信の代わりに用いた場合の実装、ハイブリッド並列実装の weak scaling 結果を示す (HA-PACS 上での実測)。XMP 記述によるオーバーヘッドが僅かに見られるが、XMP 実装の性能は概ね native な MPI 実装と遜色ないことがわかる。ユーザが逐次プログラムから並列化を行おうとする場合、MPI によるプログラミングのコストに比べ、XMP の directive 記述のみでのプログラミングは生産性を大きく向上させ、さらにハイブリッド並列では物理イメージをそのまま活かしたプログラミングが可能である。今後、アプリケーションを増やし、検証を進める。

【3】メニーコアプロセッサと CPU のハイブリッド利用による電子ダイナミクスシミュレーションの高速化 (朴)

本センターの量子物性研究部門・矢花グループでは、TDDFT (Time Dependent Density Functional Theory) に基づく電子ダイナミクスシミュレーションコード ARTED (Ab-initio Real Time Electron Dynamics simulator) を開発している。ARTED は標準的な x86 クラスタ及び京コンピュータをターゲットに MPI と OpenMP のハイブリッドプログラミングで実装されている。このコードを本センターの Intel Xeon Phi メニーコアクラスタ COMA、さらに来年度導入が予定されている最先端共同 HPC 基盤施設 (JCAHPC) での超並列メニーコアクラスタで高速化することを目指している。そこで、我々の部門と矢花グループの共同研究として ARTED を Xeon Phi 向けに最適化し、さらに CPU との併用で性能を大幅に向上させる性能チュ

ーニングを行った。

Xeon Phi は x86 アーキテクチャに基づくメニーコアプロセッサであるため、単に OpenMP コードとしてコンパイルしても動作はするが、性能特性が大幅に Xeon CPU とは異なるため、特殊なチューニングが必要である。我々の部門で、i) 配列表現の多次元化、ii) Xeon Phi 向けコンパイラオプションの調査、iii) 間接参照表現の解消等の最適化を行った結果、Xeon Phi の性能は Xeon CPU とほぼ同程度となった。理論ピーク性能比からするとこの性能は非常に低いが、この状況でも Xeon Phi と Xeon を同性能のプロセッサとして利用できれば、COMA 上のノード性能を事実上倍増することが可能である（COMA の各ノードには 2 基の Xeon CPU と 2 台の Xeon Phi カードが実装されている）。そこで、COMA で利用可能な Symmetric mode により、ノード当たり 4 つの MPI プロセスを走らせ、Xeon と Xeon Phi に 2 つずつを割り当てることにより、ノード当たり性能がほぼ倍になることを確認した。

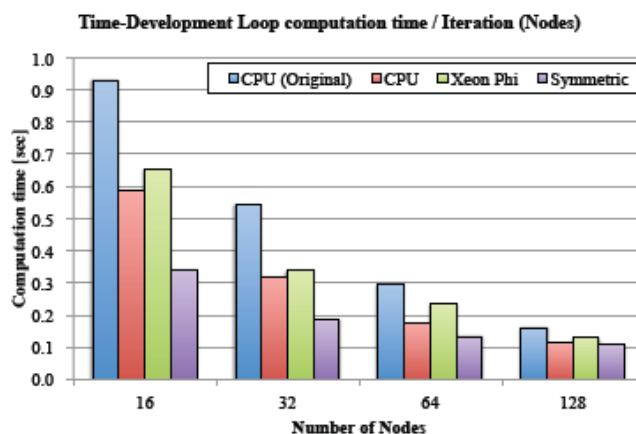


図 6-8 COMA における ARTED コードの Symmetric mode 実行の性能

図 6-8 に COMA における最大 128 ノードまでの strong scaling の性能を示す。“CPU”は Xeon 向けに最適化されたコードを CPU のみで実行させた場合の性能、“Symmetric”はノード上の CPU と Xeon Phi をフルに利用した場合を示す。16 ノードでは Symmetric の性能が CPU のほぼ倍であり、64 ノードまで良好なスケーラビリティを示している。128 ノードでは並列オーバーヘッドと、各プロセスの潜在的並列性が不足することで Xeon Phi 性能が頭打ちとなっている。しかし、両部門の共同研究の結果、ARTED の性能が大幅に加速され、今後の物性研究に供されることになった。

【4】Extreme SIMD 方式に基づく次世代演算加速機構の研究（朴、佐藤、児玉）

昨年度まで、次世代スーパーコンピュータの要素技術研究（Feasibility Study）として概

念設計及び机上性能評価により超並列型演算加速機構の研究を行ってきたが、同機構は文部科学省の次世代スーパーコンピュータには採用されず、同システムは汎用スカラープロセッサのみによるホモジニアス構成とすることが決定された。しかし、次々世代の技術として、超並列型演算加速機構が重要であることは明らかであり、特に電力当たり演算性能を最大限に引き出すためには、プロセッサのダイナミズムを最小限に抑え、電力を演算に傾注することが不可欠である。このため、筑波大学計算科学研究センター、理化学研究所計算科学研究機構、日立製作所の三者により、本演算加速機構のさらなる研究を進めることが合意され、研究を継続している。

同機構は EXACC (Extreme Accelerator) と名付けられ、超並列 SIMD アーキテクチャは Extreme-SIMD と呼ばれる。Extreme-SIMD では最大 4096 台の PE (Processing Element) の SIMD 制御を視野に、SRAM によるローカルストレージ、外部 HBM (High Bandwidth Memory) による大規模ストレージ、チップ内の 4 次元 PE 間メッシュネットワーク、チップ間の 4 次元トラスネットワーク、さらにチップ内の collective 通信を高速化する Broadcast Memory 等の様々な要素を組み込んでいる。

現状ではソフトウェアシミュレータによる性能評価、FPGA による部分モデルの検証、XcalableMP/XcalableACC コンパイラの開発が進められている。今後、詳細設計を進め、次々世代プロセステクノロジーをベースに電力性能の評価、アプリケーション開発と性能評価を進めていく予定である。

【5】メニーコアプロセッサのための並列プログラミングモデルの研究 (佐藤)

現在、筑波大学と東京大学で運営している最先端共同 HPC 基盤施設において検討を進めている Post T2K システムは、インテル Xeon Phi などのメニーコアプロセッサを多数結合したシステムになると想定されている。我々は、これからの大規模並列システムのプログラミングモデルとして、XcalableMP (XMP) の開発を進めており、メニーコア並列システムにどのように XMP を適用するか、そして、それをどのように実装設計するかについて検討した。

XMP は、基本的に分散メモリモデルのためのプログラミングモデルであり、メニーコアを 1 ノードとする並列システムには、XMP の 1 ノードをメニーコアプロセッサに割り当て、メニーコアのノード内の並列記述には OpenMP を適用することが考えられる。XMP と OpenMP の混在の記述方法についてはすでに XcalableMP 1.1 に規定されており、これに従えばよい。Xeon Phi のプログラミングモデルとしては、OpenMP が推奨されており、ノード内は OpenMP、ノード間には XMP で記述することになる。しかし、MPI と OpenMP のハイブリッドプログラミングと同じように、ノード内とノード間の 2 つのプログラミングを行わなくてはならない。

他方、一つの考え方としては、ノード内の記述も XMP で行うことにし、ノード間とノード

内を区別することなく、XMP で記述することが考えられる。XMP の通信モデルは PGAS モデルであり、共有メモリにおいては一方向通信が効率的に行うことができるため、効率的な実行が期待できる。メニーコアプロセッサでは、コアが多数あるために極力データ共有は避けたほうがよく、この点からもメリットが期待できる。但し、ノード内とノード間が均一なモデルを提供するためにステンシル計算にはいいが、ノード内とノード間を区別してプログラミングする必要がある場合には特別な配慮が必要になる。

メニーコアプロセッサのノード内にも XMP を用いるためには、ノード内の通信として、次の選択肢が考えられる。

- (1) MPI を使う。現在の XMP コンパイラが生成するオブジェクトは MPI を用いた実行時ルーチンを利用しており、これをそのまま使えばよい。しかし、ノードにおいてもプロセスで実行する必要があるため、プロセス間通信を MPI で行うため、スケーラビリティ、通信の効率に問題がある。
- (2) mmap による共有メモリの確保を行い、これを通して通信を行う。通信は、共有メモリで直接行うために高速化が期待できる。そのモデルを図 6-9 に示す。
- (3) プログラム変換によりスレッドモデルへトランスレートする。すなわち、共有データは大域変数を用い、XMP のノード内の変数はスレッドプライベートな変数に書き換える。この方法は、PGAS 言語である UPC の共有メモリ実装で行われている。
- (4) PVAS (Partitioned Virtual Address Space) モデルによる実装。PVAS は理研 AICS で開発されているメニーコア向けの新しいプロセスモデルで、一つのアドレス空間上に複数の PVAS プロセスを持つことができる。PVAS プロセスは、通常のプロセスとして用いることができるが、同じアドレス空間にある PVAS プロセス間ではお互いのメモリを直接アクセスすることができる。これを用いて、一方向通信を直接他の PVAS プロセスのデータへのアクセスで実現する。

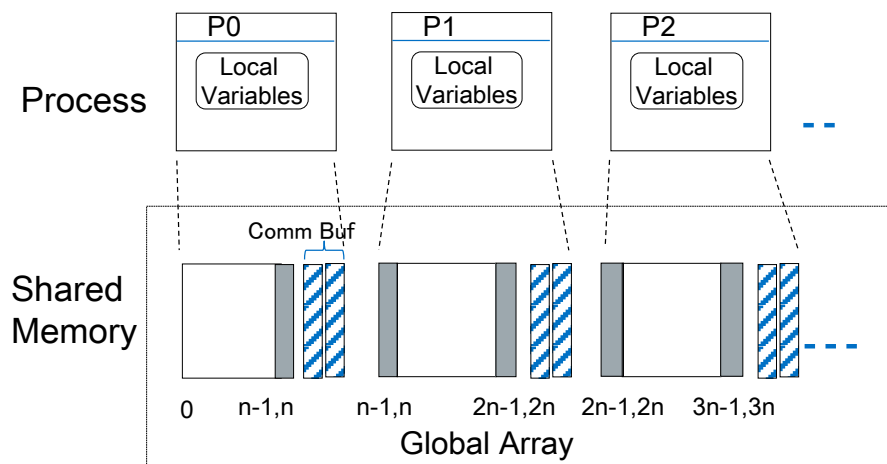


図 6-9 (2)による共有メモリ上での分散配列と局所変数の実装例 (ノード内)

以上の方法について、設計・実装し、評価を行った。(2)の方法を用いて、Himeno ベンチマークを 1 ノードで評価した結果を図 6-10 に示す。HIME_MPI と HIME_OMP は、ノード内の MPI と OpenMP の結果である。HIME_SHM は、今回新たに共有メモリによる(2)の実装の結果である。OpenMP の実装では、外側のループだけの並列化では並列性がうまく取り出せない。内側のループも含めて Collap した結果が HIME_OMP_COL であり、これにより性能を引き出すことができる。今回の実装は、これに匹敵する性能を得ることができた。

今回の実装(2)では、以下の点で性能を得ることができた。

- 共有メモリを用いることにより、メッセージ通信よりもコピー回数を削減できるために、効率化ができた。
- 各コアで XMP プロセスを動作することによって、プロセスのプライベートデータが明確に区別されるために、false share などの影響が少なくなり、プライベートデータがコアのキャッシュにとどまりやすくなり性能が向上することが確認できた。

現在、(3)、(4)についても、実装評価を進めている。

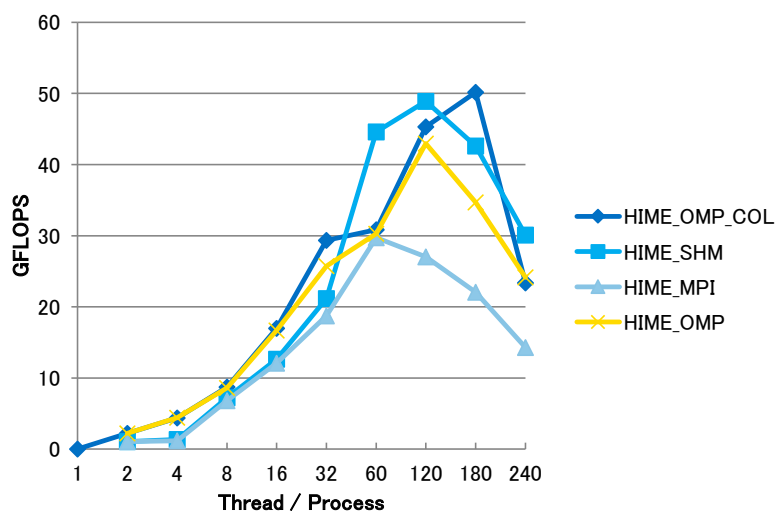


図 6-10 共有メモリを用いた XMP の実装の結果(HIME_SHM)

【6】並列高速フーリエ変換における自動チューニングの研究（高橋）

科学技術計算で広く用いられている並列高速フーリエ変換 (FFT) の性能を改善するために、自動チューニングに関する研究を行った。ポストペタスケール計算環境の一つと考えられる GPU クラスタにおいて並列 FFT を高速化する際に、最適な性能パラメータは GPU のアーキテクチャ、ノード間を結合するネットワーク、そして問題サイズなどに依存するため、これら

のパラメータをその都度手動でチューニングすることは困難になりつつある。そこで、GPU クラスタにおける並列 FFT の性能を向上させるための自動チューニング手法について検討した。GPU クラスタにおいて並列一次元 FFT を行う際には、全対全通信が 3 回行われることから、計算時間の大部分が全対全通信によって占められることになる。また、FFT を計算する際の基底や、行列の転置を行う際のブロックサイズが全体に関わる性能パラメータとなる。

GPU クラスタにおける並列一次元 FFT において、全対全通信方式、基底の組み合わせ、そしてブロックサイズについて自動チューニングを行うことで、性能をさらに向上できることが分かった。

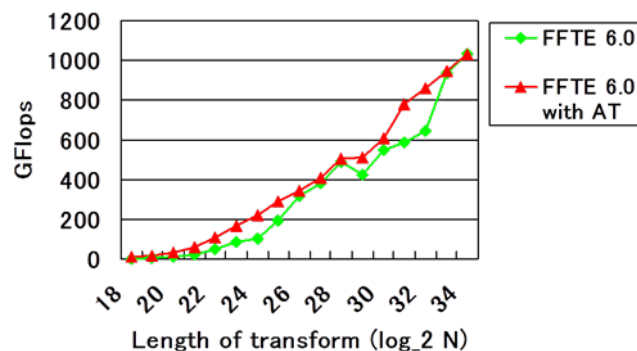


図 6-11 並列一次元 FFT の性能 (HA-PACS ベースクラスタ, 128 ノード, 512MPI プロセス)

【7】データインテンシブサイエンスのためのシステムソフトウェア (建部、川島)

本研究では分散ファイルシステム、大規模データ処理実行基盤の研究項目を実施することにより、ポストペタスケールデータインテンシブサイエンスのためのシステムソフトウェアの設計プロトタイプ実装と性能評価を行った。以下、項目別に記述する。

・分散ファイルシステム

研究の狙いは、CPU コア数の増加に対し、アクセス性能がスケールアウトし、かつアクセス応答時間が長くない分散ファイルシステムの研究開発を行うことである。本年度は、ファイルデータの冗長化について研究開発を行った。

分散ファイルシステムのアクセス性能をスケールアウトさせるために、計算ノードのローカルストレージを利用する場合、計算ノードの障害によりそのノードのローカルストレージに格納されたデータにアクセス不能となってしまう。この問題を解決するためこれまでファイル複製を他ノードのローカルストレージに保持する方法がとられたが、この方法ではストレージ領域が 2 倍以上必要となってしまう、データ転送量が 2 倍以上となってしまうという問題があった。この問題を解決するため、ノード間で冗長符号を用いることが考えられる。

しかしながらこれまではノード間での冗長符号を書き込み時に行うことは冗長符号計算のオーバーヘッドが大きく行われていなかった。そのため、書き込み時は冗長符号化のオーバーヘッドがない複製を作成し、その後、冗長符号に変換する等の方法が用いられていた。この問題を根本的に解決するため、ストレージに冗長符号計算とデータ転送をオフロードしたアクティブストレージを提案した。

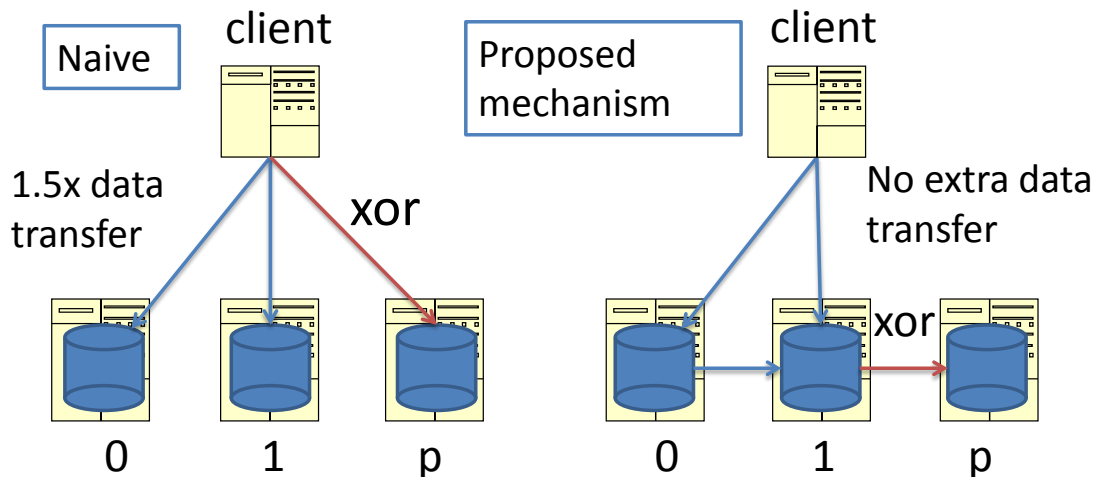


図 6-12 2D1P の RAID-4 のアクティブストレージによるデータ転送と冗長コード計算

アクティブストレージでは、冗長符号に関わる計算と通信をストレージが行うため、書込を行うクライアント側のデータ転送量、冗長符号計算のオーバーヘッドはない。2D1P (2 データディスク、1 パリティディスク) の RAID-4 の場合、図 6-12 に示すように通常の方法だとクライアントからの転送量が 1.5 倍となり、また冗長符号計算としての XOR 計算が必要となってしまう。提案のアクティブストレージの場合は、クライアントからのデータ転送量は変わらず、またクライアント側での XOR 計算は必要なくなる。その代わりに、ストレージ 0 はデータをストレージ 1 に転送し、ストレージ 1 で XOR 計算を行い、ストレージ p に転送する。本提案手法を Infiniband FDRx4 を用いた PC クラスタで初期評価を行ったところ、ほぼネットワーク物理ピーク性能となる 5.4GB/s の書込性能を達成することができた。本成果は、IEEE クラスタコンピューティングに関する国際会議でポスター発表 3)を行った。

・大規模データ処理実行基盤

本研究項目ではデータインテンシブサイエンスのアプリケーションを効率的に実行するため、MPI-I/O、大規模ワークフロー実行、MapReduce 処理、バッチキューイングシステム、データベース管理システムなどの実行環境の研究開発を行う。本研究提案で研究開発する分散ファイルシステムは、全体としてのファイルアクセス性能はスケールアウトさせるように設計、

実装を行っているが、ファイルアクセス性能が非均一である。そのため、効率的に利用するためには、ファイルアクセスの局所性の利用とデータ移動を最小化するプロセススケジューリングが重要となる。さらに、データ移動と計算処理はオーバーラップが可能であり、プロセススケジューリングとデータ移動のスケジューリングを最適化することにより、更に効率的な実行が可能となる。

大規模ワークフロー実行に関する研究では、これまでノード間のデータ移動を最小化するスケジューリング手法を提案し、評価を行ってきた。本年度は、昨年度から研究を行っている、同一ノード内での複数のジョブ実行に関して、バッファキャッシュの再利用率を向上させ、ワークフローのジョブ実行時間を短縮するスケジューリング手法の高速化を行った。昨年度からの提案手法である LIFO+HRF 手法と Rank Overlap+HRF について、天文データ解析のワークフロー Montage を用いてより詳細に評価した。LIFO+HRF、Rank Overlap+HRF の両手法は問題となっていたバッファキャッシュの有効利用が可能であり、最大既存手法と比べ 1.9 倍の高速化を達成した。さらに、末尾タスク問題も解決しており、96 コアのケースでは 12% の高速化を達成した。ノード当たりで処理するデータ量が増えるに従い、バッファキャッシュの有効利用の効果は高くなり、またコア数が増えるに従い末尾タスク問題の影響は大きくなる。今後のより大規模なデータ解析を高速化する重要なスケジューリング手法であることが分かった。

バッチキューイングシステムの研究では、引き続きデータアクセス局所性を高めるためのスケジューリング手法の研究を進めた。昨年度は I/O インテンシブなジョブを想定したスケジューリング手法を提案したが、本年度は CPU インテンシブなジョブについても扱うため、スケジューリング手法の拡張を行った。それぞれのジョブにおける入力ファイルからアクセス局所性の指標を求め、CPU の負荷と併せてパラメータ β によりノード割り当てを行う手法の設計を行った。バッチキューイングシステム Torque を元にしたプロトタイプシステムの実装を行い、BLAST ベンチマークで評価を行った。BLAST ベンチマークでは I/O インテンシブなジョブと CPU インテンシブがミックスしたベンチマークであるが、 β の値を変えることにより効率的なスケジューリングを行うことが可能となることを示した。

【8】エクストリームビッグデータの基盤技術（建部、川島）

エクストリームビッグデータ（EBD）アプリケーションの実行に求められる、数万～数十万プロセスからの並列アクセスを想定した IOPS、プロセス数に比例した読込、書込アクセスバンド幅性能を目標として、分散オブジェクトストアの設計の最適化をすすめた。

H26 年度は、H25 年度に行ったローカルオブジェクトストアの設計を元にプロトタイプ実装を行い、設計の改善を行った。flash および不揮発性メモリにおいて可能となる操作をベース

に、OpenNVM を用いてより効率的なデータアクセスが可能となるように設計を行った。基本的な設計方針は H25 年度を踏襲し、設計の最適化をすすめた。まず、オブジェクトを作成するたびにスーパーブロックを更新していたが、 n オブジェクトの作成を予約することにより、スーパーブロックの更新回数を $1/n$ に削減することが可能となる。次に、OpenNVM が提供する複数の IO ベクトルをアトミックに書き込む処理を用い、オブジェクトの初期化を、毎回オブジェクトを作成するたびに行うのではなく、 m オブジェクトを一度の IO ベクトルで一括して先行的に初期化することにより初期化処理を m 倍近く高速化することが可能となる。 $n = 128$ として、 m を変えたときのオブジェクト作成性能を図 6-13 に示す。一番下の線は $n = m = 1$ の最適化をしない昨年度のプロトタイプ実装の結果であるが、これまで 16 スレッドで 190,000 ops/sec であった。これに対し、 $n = 128$, $m = 32$ の時に 747,000 ops/sec まで性能が向上し、ほぼ 4 倍の性能向上を実現した。

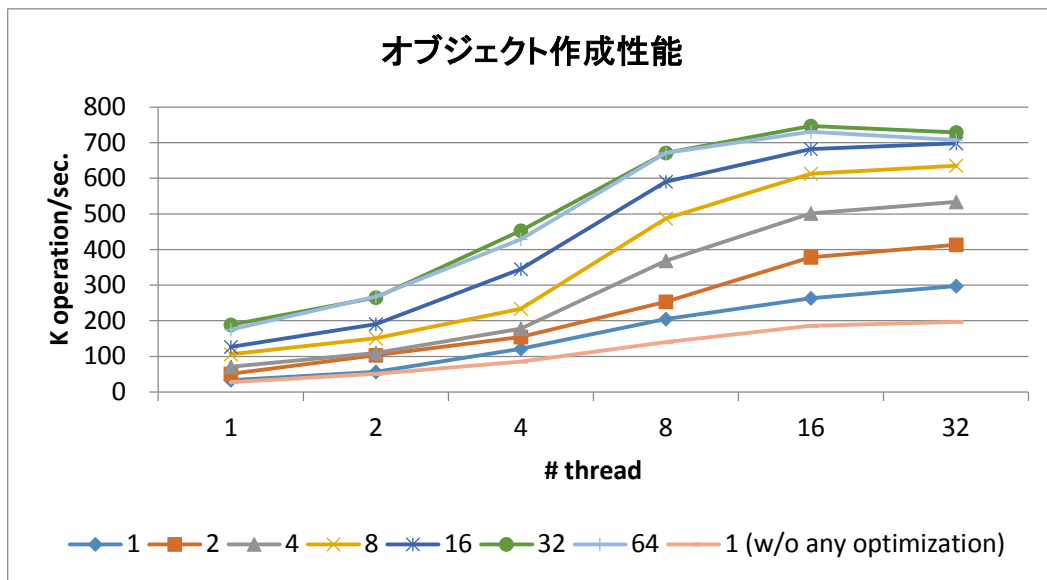


図 6-13 ローカルオブジェクトストアの作成性能の評価

並列問合せの設計においては、配列データのウィンドウパーセントイル処理において、差分計算を用いた最適化を提案した。差分計算により、ウィンドウを移動したときにその差分データについての計算だけで済むため冗長計算の削減が可能となる（図 6-14）。

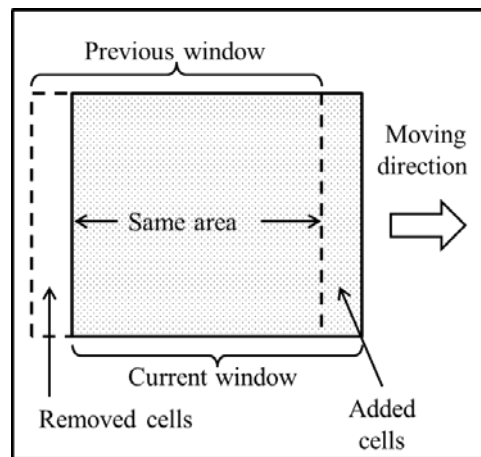


図 6-14 ウインドウアグリゲート処理における差分計算の模式図

パーセンタイル処理に応用するため、データを平衡木で保持する。これにより、気象データを用いた 30 日間のパーセンタイル処理において既存手法に比べ 13 倍の性能向上を実現した。

【9】分散ファイルシステム及びグリッド・クラウド技術に関する研究（建部）

文部科学省が進める革新的ハイパフォーマンスコンピューティングインフラ (HPCI) の HPCI 共用ストレージ、素粒子物理学データ共有システム JLDG のシステムソフトウェアとしても利用される Gfarm ファイルシステムの研究開発を行った。本年度はサイレントデータ破損 (silent data corruption) に対する対策を検討した。サイレントデータ破損とは、書込は正常に終了したものの、実際に書き込まれたデータを読み出すとデータの内容が破損していることで、ディスクコントローラの障害などが原因で引き起こされることがある。サイレントデータ破損の対策として、昨年度は書込時におけるチェックサム計算を導入した。読み込み時のチェックサム計算によりデータ破損を検知、ユーザにエラーを返す。これにより、データ破損の検知と、ユーザが破損データを読み込むことを防ぐことが可能となった。本年度は、さらにそのデータ破損の検知方法の高度化を行った。読込時のデータ破損チェックでは、データを読み込まないとデータ破損の検査を行うことができないため、データ破損の検知が遅くなる可能性がある。このため、ファイル複製作成時のデータ読込、データ書き込みにおいてデータ破損の検出を行うことができるようにプロトコルの改良を行った。Gfarm ファイルシステムでは、ファイル作成時にファイル複製を作成するため、この機構の導入によりファイル作成時のデータ破損検知が可能となる。その他、運用監視システムにおける監視項目の整備、不具合修正、ドキュメントの更新なども行った。この結果として、Gfarm バージョン 2.6.3 をリリースした。

【10】天体望遠鏡データ処理に関する研究（建部、川島）**1. 高性能な解析パイプラインシステムの設計**

すばる望遠鏡に搭載された超広視野主焦点カメラ Hyper Suprime-Cam (HSC) から得られる画像データを処理することで、研究者や一般人にとって有益な情報を有する画像を導出することができる。このデータ処理には長時間が必要である。一日分の入力データを処理するために 24 時間程度の処理時間が必要である。この処理時間を短くすることが研究の狙いである。

解析パイプラインシステムを理解するために、IPMU 側との打合せを行い、現在のパイプラインシステムのプログラムならびに入力データをご提供頂いた。その後、プログラムの実行ならびに分析を行った。パイプラインシステムを調査した結果、計算ノードからのストレージアクセスにおいて、現状は Lustre ファイルシステムを介して多数のファイルへ集中的にアクセスしていることが明らかになった。このタスクにおいて生じる集中的なアクセスが I/O ボトルネックを発生させている可能性が示唆された。そこで Gfarm 分散ファイルシステムを用いることにより、並列 I/O を活用して、このボトルネックを解消する方式を検討した。

2. 天文データ処理システムの設計

天文データ処理システムに関する研究として、二つの研究を行った。第一の研究は超新星爆発検出のためのデータ基盤に関する研究であり、第二の研究は天体カタログ上のデータ検索に関する研究である。

2.1. 超新星爆発検出のためのデータ基盤に関する研究

すばる望遠鏡から得られる画像から超新星爆発を高速に検出できるデータ基盤を開発することが本研究の目的である。検出アルゴリズムを効率的に実行できる環境を構築することが本研究の狙いである。天体画像は配列データとして表現されるため、そのデータモデルはリレーショナルデータモデルではなく配列が適切であると判断した。そこで配列データベースである SciDB を利用することにした。天体画像処理で基本となる各種集約演算が実行できることを SciDB で確認した。

また、集約演算の高性能化技法を検討した。インクリメンタル計算（図 6-15）に基づく集約演算の高性能化技法を構築し、SciDB 上に実装を行い、その性能を確かめた。具体的には配列データベースにおけるウィンドウ集約演算（最大、最小、平均、総和、パーセンタイル）を高性能化する手法を提案した。高性能化のためにインクリメンタル計算スキームを適用した。これを成功させるため、総和と平均にはリスト、最大と最小にはヒープ、パーセンタイルには二分木を用いた。提案手法を代表的な配列データベースである SciDB に実装して実験的に

評価した。その結果、提案手法は最大で従来手法に比べて 28 倍高速だった。実験に用いたコードは GitHub に公開し、成果はビッグデータにおける国際会議で報告した。

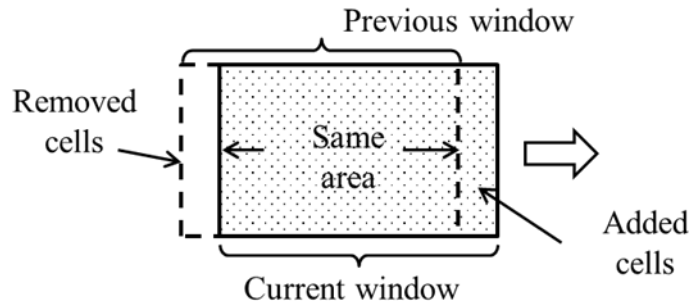


図 6-15 インクリメンタル計算スキーム

2.2. 天体カタログ上のデータ検索に関する研究

解析パイプラインから出力される結果は、天体カタログとして管理される。天体カタログの例としては SDSS が著名である。SDSS において、データはリレーショナルデータベースに登録され、SQL で問合せが行われる。本研究の狙いはこれらの問合せを高速化することである。

天体カタログでは様々な問合せ処理が行われる。今年度は IPMU との議論の結果、頻繁に使われるという近傍検索について取上げることにした。近傍検索とは、ある星の近い星を求めるという問合せに該当する。この問合せは結合演算に帰着される。この演算を高速化するために、本研究では並列分散照合処理と空間索引を組み合わせた。マシン間並列処理について cyclo-join を実装し、マシン内並列処理についてはスレッドレベル並列化と命令レベル並列化（SIMD 演算）を行った。空間索引については R 木を実装した。提案手法を評価するために実験を行った。データサイズは 100 万件とし、マシン台数は 20 とした。マシン間のネットワークには Infiniband を利用し、IPoIB を用いた。提案手法は 35 秒程度であり、ナイーブな手法に比べて、最大（選択率が 1%）で 306 倍程度の性能向上を実現した。なお、既存の著名な RDBMS である PostgreSQL で同様の実験を行ったところ 24 時間でも終了しなかったことを付記する。

【11】複数右辺ベクトルをもつ連立一次方程式の数値解法に関する研究（多田野）

複数右辺ベクトルをもつ連立一次方程式の数値解法である、Block Krylov 部分空間反復法に関する研究を行った。Block Krylov 部分空間反復法は、右辺ベクトル数が増加すると数値的に不安定な状況に陥り、残差の発散・停滞を引き起こすことがある。また、計算過程で発生する誤差の影響で、最終的に得られる近似解の精度が劣化する可能性がある。本研究では、

Block Krylov 部分空間反復法の 1 つである Block BiCGGR 法、及び Block BiCGSTAB(l) 法に対して性能改善を行った。

Block BiCGGR 法は、反復過程で現れる縦長行列を構成する列ベクトルの線形独立性が失われると数値的に不安定な状況に陥る。本研究ではこの状況を Partial Near Breakdown (PNB) procedure と呼ばれる手法を用いて検知し、線形独立性の低いベクトルを除去することで数値的安定化を図った。図 6-16 に Block BiCGGR 法、及び PNB procedure 付き Block BiCGGR 法の相対残差履歴を示す。テスト問題は数値流体力学計算で現れる連立一次方程式（行列サイズ: 13, 514、非零要素数: 352, 762、右辺ベクトル数: 32）である。ここで、 δ は PNB procedure を制御するパラメータである。Block BiCGGR 法の相対残差は反復序盤に発散したが、PNB procedure を用いることで Block BiCGGR 法の相対残差は収束した。

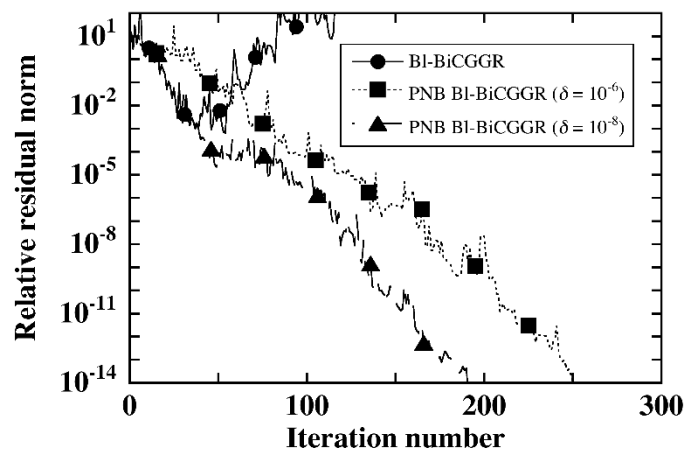


図 6-16 Block BiCGGR 法、及び PNB procedure 付き Block BiCGGR 法の相対残差履歴

Block BiCGSTAB(l) 法においても、反復過程で現れる縦長行列を構成するベクトルの線形独立性が失われると数値的に不安定な状況に陥る。本研究では反復毎に正規直交化を実行することで数値的不安定性を回避した。また同法は反復過程で発生する誤差の影響で最終的に得られる近似解の精度が劣化するが、近似解の精度を保つために行列ベクトル積計算を追加し、高精度化を図った。図 6-17 に Block BiCGSTAB(l) 法、及び同法に正規直交化、高精度化を行ったときの真の相対残差履歴を示す。テスト問題は数値流体力学計算で現れる連立一次方程式（行列サイズ: 23, 560、非零要素数: 460, 598、右辺ベクトル数: 16）である。Block BiCGSTAB(l) 法のパラメータとして $l = 8$ を用いた。Block BiCGSTAB(l) 法を適用した場合は相対残差が発散したが、反復過程で正規直交化を実行することで相対残差が収束した。しかしながら、真の相対残差は 10^{-5} 付近で停滞しており、高精度の近似解は得られていない。一方、正規直交化に加えて高精度化の処理を追加することにより真の相対残差は 10^{-11} 付近まで減少しており、高精度の近似解を生成することができた。

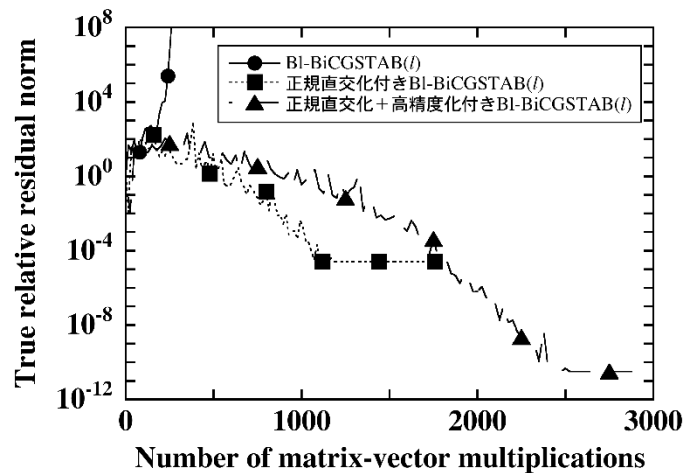


図 6-17 Block BiCGSTAB(l)法、正規直交化付き Block BiCGSTAB(l)法、及び正規直交化+高精度化付き Block BiCGSTAB(l)法の真の相対残差履歴

4. 教育

【学生の指導状況】

1. 藤井久史、修士（工学）、GPU 間直接通信機構に基づく並列アプリケーションの性能評価、筑波大学大学院システム情報工学研究科修士論文、平成 27 年 3 月（指導：朴泰祐）
2. 田渕晶大、修士（工学）、GPU クラスタ向けプログラミングモデル XcalableACC のコンパイラと通信最適化、筑波大学大学院システム情報工学研究科修士論文、平成 27 年 3 月（指導：佐藤三久）
3. 津金佳祐、修士（工学）、PGAS 言語 XcalableMP による並列アプリケーションの実装と評価、筑波大学大学院システム情報工学研究科修士論文、平成 27 年 3 月（指導：佐藤三久）
4. 平櫛貴章、修士（工学）、GPU を用いた並列および分散幅優先探索アルゴリズムの高速化、筑波大学大学院システム情報工学研究科修士論文、平成 27 年 3 月（指導：高橋大介）
5. キョウユ、修士（工学）、分散ファイルシステム Gfarm における MTC アプリケーションの性能予測モデルの構築、筑波大学大学院システム情報工学研究科修士論文、平成 27 年 3 月（指導：建部修見）

【集中講義】

1. 「CCS HPC サマーセミナー」(一般公開、全学共通科目)、2014/9/1～9/2
2. “HPC Autumn Seminar”(全学共通科目) 2014/9/16～9/17

5. 受賞、外部資金、知的財産権等

【受賞】

1. Best Paper Award in Fifth International Symposium on Highly-Efficient Accelerators and Reconfigurable Technologies (HEART2014), Y. Kodama, T. Hanawa, T. Boku and M. Sato, “PEACH2: An FPGA-based PCIe network device for Tightly Coupled Accelerators”, 2014/6/9.
2. HPC Challenge Class2 Best Performance Award at SC14, Masahiro Nakao, Hitoshi Murai, Hidetoshi Iwashita, Takenori Shimosaka, Akihiro Tabuchi, Taisuke Boku, Mitsuhsa Sato, “XcalableMP and XcalableACC for Productivity and Performance”, 2014/11/18.
3. Best Poster Awards in HPC in Asia Poster Session, ISC2014, T. Hanawa, Y. Kodama, T. Boku, M. Sato, “Proprietary Interconnect with Low Latency for HA-PACS/TCA”, 2014/6/26.
4. 情報科学技術フォーラム(FIT2014) 船井ベストペーパー賞, 丸山裕士, 山口佳樹, 児玉祐悦, 実時間動画像ブレ補正システムにおける補正精度向上に関する研究, 2014 年 9 月 3 日

【外部資金】

1. JST CREST、朴泰祐(研究代表者)、H24～29 年度、22,670 千円(H26 年度)、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」、「ポストペタスケール時代に向けた演算加速機構・通信機構統合環境の研究開発」
2. JST CREST、高橋大介(共同研究者)、H24～28 年度、15,132 千円(H26 年度)、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」、「数値計算ライブラリによる超並列複合システムの階層的抽象化に関する研究」
3. 科学研究費補助金 新学術領域研究(研究領域提案型)、高橋大介(分担)、H22～26 年度、3,510 千円(H26 年度)、「大規模並列環境における数値計算アルゴリズム」
4. 科学研究費補助金 基盤研究(C)、高橋大介(代表)、H24～26 年度、1,430 千円(H26 年度)、「エクサスケール計算環境に向けた高速フーリエ変換のアルゴリズムに関する研究」
5. JST CREST、建部修見(研究代表者)、H23 年度～H27 年度、36,579 千円(H26 年度)、

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」、
「ポストペタスケールデータインテンシブサイエンスのためのシステムソフトウェア」

6. JST CREST、建部修見（共同研究者）、H25～30 年度、5,044 千円（H26 年度）、研究領域「ビッグデータ統合利活用のための次世代基盤技術の創出・体系化」、「EBD：次世代の年ヨッタバイト処理に向けたエクストリームビッグデータの基盤技術」
7. KDDI 研究所共同研究、川島英之（代表）、H26 年度、1,000 千円、「データ仮想化システムにおける大規模データ処理方式に関する研究」
8. JST CREST、川島英之（分担）、H26 年度 1,000 千円、研究領域「科学的発見・社会的課題解決に向けた各分野のビッグデータ利活用推進のための次世代アプリケーション技術の創出・高度化」、「広域撮像探査観測のビッグデータ分析による統計計算宇宙物理学」
9. JST A-STEP、川島英之（代表）、H26 年度 1,000 千円、「暗号化データ処理基盤システムの研究」

6. 研究業績

(1) 研究論文

A) 査読付き論文

1. M. Noda, K. Ishimura, K. Nobusada, K. Yabana, T. Boku: Massively-parallel electron dynamics calculations in real-time and real-space: Toward applications to nanostructures of more than ten-nanometers in size, Journal of Computational Physics, Vol.265, pp.145-155, 2014.
2. Daichi Mukunoki, Toshiyuki Imamura and Daisuke Takahashi: Fast Implementation of General Matrix-Vector Multiplication (GEMV) on Kepler GPUs, Proc. 23rd Euromicro International Conference on Parallel, Distributed and Network-based Processing (PDP 2015), pp. 642--650 (2015).
3. 佐藤佳州, 高橋大介: 対局に基づいた教師データの重要度の学習, 情報処理学会論文誌, Vol. 55, No. 11, pp. 2399-2409 (2014).
4. Yukihiro Hasegawa, Jun-Ichi Iwata, Miwako Tsuji, Daisuke Takahashi, Atsushi Oshiyama, Kazuo Minami, Taisuke Boku, Hikaru Inoue, Yoshito Kitazawa, Ikuo Miyoshi and Mitsuo Yokokawa: Performance evaluation of ultra-large-scale first-principles electronic structure calculation code on the K computer, International

- Journal of High Performance Computing Applications, Vol. 28, No. 3, pp. 335--355 (2014).
5. Yutaka Maruyama, Norio Yoshida, Hiroto Tadano, Daisuke Takahashi, Mitsuhisa Sato and Fumio Hirata: Massively parallel implementation of 3D-RISM calculation with volumetric 3D-FFT, Journal of Computational Chemistry, Vol. 35, No. 18, pp. 1347--1355 (2014).
 6. Satoshi Matsuoka, Hitoshi Sato, Osamu Tatebe, Michihiro Koibuchi, Ikki Fujiwara, Shuji Suzuki, Masanori Kakuta, Takashi Ishida, Yutaka Akiyama, Toyotaro Suzumura, Koji Ueno, Hiroki Kanezashi, Takemasa Miyoshi, "Extreme Big Data (EBD): Next Generation Big Data Infrastructure Technologies Towards Yottabyte/Year", Supercomputing Frontiers and Innovations, Vol. 1, No. 2, pp.89-107, 2014 (DOI: <http://dx.doi.org/10.14529/jsfi140206>)
 7. Yoshihiro Oyama, Shun Ishiguro, Jun Murakami, Shin Sasaki, Ryo Matsumiya, Osamu Tatebe, "Reduction of Operating System Jitter Caused by Page Reclaim", In Proceedings of the 4th International Workshop on Runtime and Operating Systems for Supercomputers (ROSS 2014), Munich, Germany, June 10, 2014. (DOI: 10.1145/2612262.2612270)
 8. Masahiro Tanaka, Osamu Tatebe, "Disk Cache-Aware Task Scheduling For Data-Intensive and Many-Task Workflow", Proceedings of IEEE International Conference on Cluster Computing (Cluster), pp.167-175, 2014 (DOI: 10.1109/CLUSTER.2014.6968774)
 9. Hiroki Ohtsuji, Osamu Tatebe, "POSTER: Preliminary evaluation of optimized transfer method for cluster-wide RAID-4", Proceedings of IEEE International Conference on Cluster Computing (Cluster), pp.284-285, 2014 (DOI: 10.1109/CLUSTER.2014.6968775)
 10. Jiang Li, Hideyuki Kawashima, Osamu Tatebe, "Incremental Window Aggregates over Array Database", Proceedings of IEEE International Conference on Big Data (BigData), short paper, pp.183-188, 2014
 11. Y. Maruyama, N. Yoshida, H. Tadano, D. Takahashi, M. Sato, and F. Hirata, Massively parallel implementation of 3D-RISM calculation with volumetric 3D-FFT.

12. H. Tadano, Y. Ishikawa, A. Imakura, Improvement of the accuracy of the approximate solution of the Block BiCR method, JSIAM Letters, Vol. 6, pp. 61—64, 2014.
13. S. Saito, H. Tadano, and A. Imakura, Development of the Block BiCGSTAB(l) method for solving linear systems with multiple right-hand sides, JSIAM Letters, Vol. 6, pp. 65—68, 2014.

B) 査読無し論文

(2) 国際会議発表

A) 招待講演

1. Taisuke Boku, “Tightly Coupled Parallel Accelerated Computing for Post-Petascale Computing”, Invited Talk, French Japan Conference, Tokyo, Apr. 14th, 2014.
2. Taisuke Boku, “Towards Reconfigurable High Performance Computing based on Co-Design Concept”, Keynote Talk, Fifth International Symposium on Highly-Efficient Accelerators and Reconfigurable Technologies (HEART2014), 2014/6.
3. Taisuke Boku, “Next Generation Parallel Accelerated Computing with High Productivity”, Keynote Talk, Int. Workshop on Enhancing Parallel Scientific Applications with Accelerated HPC (ESAA 2014) in EuroMPI/Asia 2014, Kyoto, Sept. 9th, 2014.
4. Taisuke Boku, “Accelerator and Communication CoDesign Challenges at Extreme Scale”, Invited Talk, CODESIGN2014, Guangzhou, Nov. 7th, 2014.
5. Taisuke Boku, “Research and Development on Unified Environment of Accelerated Computing and Interconnection for Post-Petascale Era”, Invited Talk, 2014 ATIP Workshop on Japanese Research Toward Next-Generation Extreme Computing (with SC14), New Orleans, Nov. 17th, 2014.
6. Osamu Tatebe, “System Software for Post-Petascale Data-Intensive Science”, Invited Talk, 2014 ATIP Workshop on Japanese Research Toward Next-Generation Extreme Computing (with SC14), New Orleans, Nov. 17th, 2014.
7. Taisuke Boku, “Tightly Coupled Accelerators for Accelerator-oriented Parallel Processing”, Invited Talk, JST/CREST International Symposium on Post Petascale System Software (ISP2S2 2014), Kobe, Dec. 4th, 2014.

8. Osamu Tatebe, “System software for Post-Petascale Data-Intensive Science”,
Invited Talk , JST/CREST International Symposium on Post Petascale System
Software (ISP2S2 2014), Kobe, Dec. 3rd, 2014.

B) 一般講演

1. K. Tsugane, H. Nuga, T. Boku, H. Murai, M. Sato, W. Tang, B. Wang, “Hybrid-view Programming of Nuclear Fusion Simulation Code in the PGAS Parallel Programming Language XcalableMP”, Proc. of ICPADS2014, Hsinchu, 2014.
2. Masahiro Nakao, Hitoshi Murai, Takenori Shimosaka, Akihiro Tabuchi, Toshihiro Hanawa, Yuetsu Kodama, Taisuke Boku, Mitsuhsa Sato.
“XcalableACC: Extension of XcalableMP PGAS Language using OpenACC for Accelerator Clusters”, Workshop on accelerator programming using directives (WACCPD), New Orleans, LA, USA, Nov., 2014.
3. N. Fujita, H. Fujii, T. Hanawa, Y. Kodama, T. Boku, Y. Kuramashi, M. Clark,
“QCD Library for GPU Cluster with Proprietary Interconnect for GPU Direct Communication”, Proc. of HeteroPar 2014 (with EuroPar 2014), Porto, 2014.
4. N. Fujita, H. Nuga, T. Boku, Y. Idomura, “Nuclear Fusion Simulation Code Optimization and Performance Evaluation on GPU Clusters”, Proc. of PDSEC2014 (with IPDPS2014), Phoenix, 2014.
5. Daisuke Takahashi: Automatic Tuning for Parallel FFTs on GPU Clusters, 2015 SIAM Conference on Computational Science and Engineering (CSE15), Salt Palace Convention Center, Salt Lake City, Utah, USA, March 18, 2015.
6. Hiroshi Maeda and Daisuke Takahashi: Performance Evaluation of Sparse Matrix-Vector Multiplication Using GPU/MIC Cluster, 2015 SIAM Conference on Computational Science and Engineering (CSE15), Salt Palace Convention Center, Salt Lake City, Utah, USA, March 14, 2015.
7. Daisuke Takahashi: Automatic Tuning for Parallel FFTs on GPU Clusters, 2015 Conference on Advanced Topics and Auto Tuning in High-Performance and Scientific Computing (2015 ATAT in HPSC), National Taiwan University, Taipei, Taiwan, February 28, 2015.
8. Y. Kodama, T. Hanawa, T. Boku and M. Sato, “PEACH2: An FPGA-based PCIe network device for Tightly Coupled Accelerators”, Fifth International Symposium

- on Highly-Efficient Accelerators and Reconfigurable Technologies (HEART2014), 2014/6.
9. Yoshihiro Oyama, Shun Ishiguro, Jun Murakami, Shin Sasaki, Ryo Matsumiya, Osamu Tatebe. “Reduction of Operating System Jitter Caused by Page Reclaim”, the 4th International Workshop on Runtime and Operating Systems for Supercomputers (ROSS 2014), Munich, Germany, June 10, 2014
 10. Masahiro Tanaka, Osamu Tatebe, “Disk Cache-Aware Task Scheduling For Data-Intensive and Many-Task Workflow”, IEEE International Conference on Cluster Computing (Cluster), Madrid, Spain, September 24, 2014
 11. Jiang Li, Hideyuki Kawashima, Osamu Tatebe, “Incremental Window Aggregates over Array Database”, IEEE International Conference on Big Data (BigData), Washington D.C., October 30, 2014
 12. Osamu Tatebe, “System Software for Data-Intensive Science”, ICT International Exchange Workshop 2014, Tsukuba, Nov 1, 2014
 13. Osamu Tatebe, “System Software for Post-Petascale Data-Intensive Science”, 2014 ATIP Workshop: Japanese Research Toward Next-Generation Extreme Computing, New Orleans, Nov 17, 2014
 14. Osamu Tatebe, “System software for Post-Petascale Data-Intensive Science”, JST/CREST International Symposium on Post Petascale System Software (ISP2S2 2014), Kobe, Japan, December 3, 2014
 15. Masahiro Tanaka, “Workflow system for data-intensive many-task computing”, JST/CREST International Symposium on Post Petascale System Software (ISP2S2 2014), Kobe, Japan, December 4, 2014
 16. Osamu Tatebe, “System Software for Post-Petascale Data-Intensive Science”, 10th AEARU Workshop on Computer Science and Web Technology, Tsukuba, Feb 27, 2015
 17. H. Tadano, “Stabilization of the Block BiCGGR method by the partial near breakdown procedure”, International Conference on Simulation Technology (JSST2014), Kitakyushu, Japan, Oct. 2014.
 18. H. Jin, Y. Yamaguchi and Y. Kodama, “Efficient shared cache system for ARM-based multi-core processors”, Fifth International Symposium on Highly-Efficient Accelerators and Reconfigurable Technologies (HEART2014), poster, 2014/6

19. T. Hanawa, Y. Kodama, T. Boku, M. Sato, “Tightly Coupled Accelerators Architecture for Low-Latency Inter-node Communication between Accelerators”, SC14 Poster Session, New Orleans, 2014.
20. T. Hanawa, Y. Kodama, T. Boku, M. Sato, “Proprietary Interconnect with Low Latency for HA-PACS/TCA”, HPC in Asia Session (poster) in Int. Supercomputing Conference (ISC) 2014, Leipzig, 2014.

(3) 国内学会・研究会発表

A) 招待講演

1. 建部 修見, “不揮発性メモリ向けファイルシステムの設計”, MemoryPlus Workshop, 東京, 2014 年 9 月 17 日
2. 川島英之, “機械学習・データマイニングのツール紹介”, 電子情報通信学会 ASN 研究チュートリアルセッション, 2015 年 3 月 11 日.
3. 齋藤周作, 多田野寛人, 今倉暁, “Block BiCGSTAB(1)法の計算量削減”, 日本応用数理学会 2014 年度年会, 政策研究大学院大学, 2014 年 9 月 3 日-5 日.
4. 朴泰祐, “大学スパコンセンターにおけるシステム導入とその役割”, FIT2014 企画「エクサスケールコンピューティング時代の大学スパコンセンターの役割」, 2014 年 9 月 3 日.
5. 朴泰祐, “高性能計算技術と計算宇宙物理～これまでとこれから～”, 第 27 回理論懇シンポジウム, 2014 年 12 月 25 日.

B) その他の発表

1. 藤井久史, 塙敏博, 児玉祐悦, 朴泰祐, 佐藤三久, “GPU 向け FFT コードの TCA アーキテクチャによる実装と性能評価”, 情報処理学会研究報告ハイパフォーマンスコМП्यूティング研究会, 2015-HPC-148, 2015.
2. 廣川祐太, 朴泰祐, 佐藤駿丞, 矢花一浩, “実時間実空間密度汎関数理論による電子動力学シミュレーションの Xeon Phi クラスタ向け最適化”, 情報処理学会研究報告ハイパフォーマンスコМП्यूティング研究会, 2015-HPC-148, 2015.
3. 桑原悠太, 塙敏博, 児玉祐悦, 朴泰祐, “GPU クラスタにおける GPU 間セルフ通信機構に関する提案”, 情報処理学会研究報告ハイパフォーマンスコМП्यूティング研究会, 2015-HPC-148, 2015.

4. 中尾昌広, 村井均, 岩下英俊, 下坂健則, 朴泰祐, 佐藤三久, “PGAS 言語 XcalableMP を用いた HPC Challenge ベンチマークの実装と評価”, 情報処理学会研究報告ハイパフォーマンส์コンピューティング研究会, 2015-HPC-148, 2015.
5. 宇川斉志, 佐藤三久, 朴泰祐, 児玉祐悦, 山口佳樹, 山本淳二, “Extreme SIMD アーキテクチャのプログラミングモデル拡張 C による性能評価”, 情報処理学会研究報告ハイパフォーマンส์コンピューティング研究会, 2015-HPC-148, 2015.
6. 松本和也, 塙敏博, 児玉祐悦, 藤井久史, 朴泰祐, “密結合並列演算加速機構 TCA を用いた GPU 間直接通信による Collective 通信の実装と予備評価”, 情報処理学会研究報告ハイパフォーマンส์コンピューティング研究会, 2014-HPC-147, 2014.
7. 小田嶋哲哉, 塙敏博, 児玉祐悦, 朴泰祐, 村井均, 中尾昌広, 佐藤三久, “HA-PACS/TCA における TCA および InniBand?ハイブリッド通信”, 情報処理学会研究報告ハイパフォーマンส์コンピューティング研究会, 2014-HPC-147, 2014.
8. 藤井久史, 藤田典久, 塙敏博, 児玉祐悦, 朴泰祐, 佐藤三久, 藏増 嘉伸, Mike Clark, “GPU 向け QCD ライブラリ QUDA の TCA アーキテクチャ実装の性能評価”, 情報処理学会研究報告ハイパフォーマンส์コンピューティング研究会, 2014-HPC-145, 2014.
9. 津金 佳祐, 奴賀 秀男, 朴 泰祐, 村井 均, 佐藤 三久, William Tang. “並列言語 XcalableMP による核融合シミュレーションコードの実装と評価”, 情報処理学会研究報告ハイパフォーマンส์コンピューティング研究会, 2014-HPC-145, 2014.
10. 田渕 晶大, 村井 均, 朴 泰祐, 佐藤 三久, “XcalableMP と OpenACC の統合による GPU クラスタ向け並列プログラミングモデル”, 情報処理学会研究報告ハイパフォーマンส์コンピューティング研究会, 2014-HPC-145, 2014.
11. 宮島 敬明, 久原 拓也, 塙 敏博, 児玉 祐悦, 朴 泰祐, 天野 英晴, “PEACH3 の基本転送性能の予備評価”, 情報処理学会研究報告計算機アーキテクチャ研究会, 2014-ARC-203, 2014.
12. 松本和也, 塙敏博, 児玉祐悦, 藤井久史, 朴泰祐, “密結合並列演算加速機構 TCA を用いた GPU 間直接通信による CG 法の実装と予備評価”, 情報処理学会研究報告ハイパフォーマンส์コンピューティング研究会, 2014-HPC-144, 2014.
13. 丸山裕士, 山口佳樹, 児玉祐悦, “実時間動画像ブレ補正システムにおける補正精度向上に関する研究”, 情報科学技術フォーラム(FIT2014), RC-004, pp. 23-28, 2014/9

14. 丸山裕士, 李成哲, 山口佳樹, 児玉祐悦, “バレーボール試合映像からの実時間選手追跡システム”, 情報科学技術フォーラム(FIT2014), C-021, pp. 225-226, 2014/9
15. 藤浪健太, 山口佳樹, 児玉祐悦, “FPGA による三次元格子ガスオートマトン法の実装”, 第 28 回数値流体力学シンポジウム, (E07-2), pp. 1-4, 2014/12
16. 伊藤剛浩, 山口佳樹, 児玉祐悦, 山本淳二, 中川八穂子, 朴泰祐, 佐藤三久, “FPGA を用いた ExtremeSIMD アーキテクチャの予備評価”, 電子情報通信学会 2015 年総合大会講演論文集, (D-6-9), 1page, 2015/3
17. 高山尋考, 藤浪健太, 山口佳樹, 児玉祐悦, “マルチ FPGA による流体シミュレーションの拡張”, 電子情報通信学会 2015 年総合大会講演論文集, (D-18-5), 1page, 2015/3
18. 高橋大介: GPU クラスタにおける並列 FFT の自動チューニング, 計算工学講演会論文集, Vol. 19, F-7-3 (2014).
19. 前田広志, 高橋大介: GPU/MIC クラスタにおける疎行列ベクトル積の性能評価, 情報処理学会研究報告, Vol. 2014-HPC-144, No. 4 (2014).
20. Li Jiang, Hideyuki Kawashima, Osamu Tatebe, “Implementing Incremental Aggregate Computation on SciDB”, 情報処理学会システムソフトウェアとオペレーティング・システム (OS) 研究発表会, 朱鷺メッセ, 2014 年 7 月 28 日
21. 川島英之, 建部修見, 分析用データ処理系における効率的なデータ配送機構, 情報処理学会システムソフトウェアとオペレーティング・システム (OS) 研究発表会, 朱鷺メッセ, 2014 年 7 月 29 日
22. 大辻弘貴, 建部修見, Cluster-wide RAID の実装と評価, 情報処理学会ハイパフォーマンส์コンピューティング (HPC) 研究発表会, 朱鷺メッセ, 2014 年 7 月 30 日
23. 鷹津冬将, 建部修見, WebGfarm: 分散ファイルシステム Gfarm の HTTP インタフェース, 情報処理学会ハイパフォーマンส์コンピューティング (HPC) 研究発表会, 朱鷺メッセ, 2014 年 7 月 30 日
24. 大辻弘貴, 建部修見, Cluster-wide RAID 向けの集中型コントローラ, 情報処理学会ハイパフォーマンส์コンピューティング (HPC) 研究発表会, 小樽, 2014 年 12 月 9 日
25. キョウユ, 建部修見, 田中昌宏, 分散ファイルシステム Gfarm における MTC アプリケーションの性能予測モデルの構築, 情報処理学会ハイパフォーマンส์コンピューティング (HPC) 研究発表会, 小樽, 2014 年 12 月 9 日

26. 島貫稔之、川島英之：シーケンス演算子の効率的実装と適応的处理機構，コンピュータセキュリティシンポジウム 2014, Vol. 2014, No. 2, pp. 976-983.

(4) 著書、解説記事等

7. 異分野間連携・国際連携・国際活動等

【国際スクール開催】

1. Korea-Japan HPC Joint Winter School, Seoul, Korea, 2015/1/20～1/2

8. シンポジウム、研究会、スクール等の開催実績

1. 第 2 回 XMP ワークショップ：PC クラスタコンソーシアム XcalableMP 分科会（主査：佐藤三久）と JST-CREST 研究課題（研究代表者：朴泰祐）による共催，東京，2015 年 10 月 25 日．

9. 管理・運営

1. 高橋大介：筑波大学情報環境機構学術情報メディアセンター運営委員
2. 朴泰祐：筑波大学情報環境機構企画室委員

10. 社会貢献・国際貢献

1. Taisuke Boku: Organizing Chair, HPC in Asia Session, ISC2015
2. Taisuke Boku: Poster Chair, SC14
3. Taisuke Boku: Progmra Committee, System Software Area, SC14
4. Taisuke Boku: Committee, 2014 ACM Gordon Bell Prize Selection Committee
5. Taisuke Boku: Program Committee, HiPC 2014
6. Taisuke Boku: Program Committee, CCGrid2015
7. Yetsu Kodama: Program Committee, IEEE Symposium on Low-Power and High-Speed Chips (Cool Chips) VXII
8. Yetsu Kodama: Program Committee, International Conference on Parallel Processing 2015
9. Yetsu Kodama: Program Committee, The 2015 4th ICT International Student Project Conference (ICT-ISPC 2015)
10. Yetsu Kodama: Reviewer, IEEE Micro Magazine
11. Osamu Tatebe: Program Committee, 14th IEEE/ACM International Symposium on Cluster, Cloud and Grid Computing (CCGrid) 2014

12. Osamu Tatebe: Program Committee, IEEE International Conference on Cluster Computing (Cluster) 2014
13. Osamu Tatebe: Program Committee, 6th IEEE International Conference on Cloud Computing Technology and Science (CloudCom) 2014
14. Osamu Tatebe: Program Committee, 20th IEEE International Conference on Parallel and Distributed Systems (ICPADS) 2014
15. Osamu Tatebe: Technical Poster Committee, 2014 ACM/IEEE International Conference on High Performance Computing, Networking, Storage and Analysis (SC14)
16. Osamu Tatebe: Program Committee, 43rd International Conference on Parallel Processing (ICPP) 2014
17. Osamu Tatebe: Program Committee, 23rd International Conference on Computer Communications and Networks (ICCCN) 2014
18. Osamu Tatebe: Program Committee, International Supercomputing Conference (ISC) 2014
19. Osamu Tatebe: Program Committee, 4th International Workshop on Advances in High-Performance Computational Earth Sciences: Applications & Frameworks (IHPCES) 2014
20. Osamu Tatebe: Scientific Committee, 11th International Meeting on High-Performance Computing for Computational Science (VECPAR) 2014
21. Osamu Tatebe: Program Committee, Architecture, Languages, Compilation and Hardware support for Emerging ManYcore systems (ALCHEMY) Workshop 2015
22. Osamu Tatebe: Program Committee, 3rd Workshop on Big Data Management in Clouds (BigDataCloud) 2014
23. Hideyuki Kawashima: Program Committee, 5th International Workshop on Streaming Media Delivery and Management Systems, In Conjunction with 9th 3PGCIC-2013 Conference

24. Hideyuki Kawashima: Program Committee, The IEEE International Conference on Multimedia Big Data (BigMM).
25. 朴泰祐: 情報処理学会 Annual Meeting on Advanced Computing System and Infrastructure (ACSI) 2015, 組織委員長
26. 児玉祐悦: 情報処理学会 デジタルプラクティス査読委員
27. 児玉祐悦: 電子情報通信学会 論文誌査読委員
28. 児玉祐悦: 電子情報通信学会 論文誌特集号「Low-Power and High-Speed Chips」編集員
29. 児玉祐悦: Annual Meeting on Advanced Computing System and Infrastructure (ACSI) 2015, 広報委員長
30. 高橋大介: 情報処理学会ハイパフォーマンスコンピューティング研究会運営委員
31. 高橋大介: 情報処理学会論文誌編集委員会委員
32. 高橋大介: Annual Meeting on Advanced Computing System and Infrastructure (ACSI) 2015 プログラム委員
33. 建部修見: HPCI 連携サービス運営・作業部会委員
34. 建部修見: 理化学研究所客員研究員
35. 建部修見: HPCI 利用研究課題審査委員会レビューアー
36. 建部修見: 学際大規模情報基盤共同利用・共同研究拠点 (JHPCN) 課題審査委員
37. 建部修見: 東京工業大学学術国際情報センター共同利用専門委員
38. 建部修見: 特定非営利団体つくば OSS 技術支援センター理事長
39. 建部修見: 情報処理学会ハイパフォーマンスコンピューティング研究会運営委員
40. 建部修見: Annual Meeting on Advanced Computing System and Infrastructure (ACSI) 2015 プログラム委員
41. 建部修見: インターネットカンファレンス 2014 プログラム委員
42. 川島英之: 情報処理学会 データベースシステム研究会 幹事
43. 川島英之: 情報処理学会論文誌編集委員会委員
44. 川島英之: 電子情報通信学会 知的環境とセンサネットワーク研究会 運営委員
45. 川島英之: 電子情報通信学会英文論文誌小特集号「Special Section on Emerging Technologies on Ambient Sensor Networks Toward Future Generation」編集委員

46. 多田野寛人：日本応用数理学会「行列・固有値問題の解法とその応用」研究部会
幹事
47. 多田野寛人：日本応用数理学会「若手の会」研究部会 運営委員
48. 多田野寛人：日本応用数理学会「応用数理」編集委員
49. 多田野寛人：日本応用数理学会 JSIAM Letters 編集委員
50. 多田野寛人：情報処理学会ハイパフォーマンスコンピューティングと計算科学シン
ポジウム HPCS2015 実行委員
51. 多田野寛人：情報処理学会ハイパフォーマンスコンピューティングと計算科学シン
ポジウム HPCS2015 プログラム委員
52. 多田野寛人：情報処理学会 Annual Meeting on Advanced Computing System and
Infrastructure (ACSI) 2015 組織委員
53. 多田野寛人：情報処理学会 Annual Meeting on Advanced Computing System and
Infrastructure (ACSI) 2015 プログラム委員
54. 朴泰祐：東京大学情報基盤センター・スーパーコンピュータ部門専門委員
55. 朴泰祐：理化学研究所計算科学研究機構・運営諮問委員

11. その他