様　式　Ｃ－１９、Ｆ－１９－１、Ｚ－１９　（共通）

科学研究費助成事業　　研究成果報告書

令和　５ 年　 ６ 月　 ３ 日現在

機関番号：１４４０１
研究種目：基盤研究(C)（一般）
研究期間：2017～2022
課題番号：１７Ｋ００１６８

研究課題名（和文）OpenFlow結合網配備クラスタを対象としたMPI実行時計算・通信連携機構

研究課題名（英文）A Coordination Structure for MPI Execution and Communication on OpenFlow-based Cluster System

研究代表者
伊達　進（Susumu, Date）

大阪大学・サイバーメディアセンター・准教授

研究者番号：２０３４６１７５

交付決定額（研究期間全体）：（直接経費）　　3,600,000円

研究成果の概要（和文）：本研究では、ネットワークは制御できない静的な資源であるという前提から、動的に制御できる資源であるという視点に基づき、クラスタシステムのインターコネクト（相互結合網)を動的に制御可能なネットワーク資源ととらえ、MPIプログラム特性とSDNのネットワークプログラミング性をシームレスに連携させるという新しいアプローチで、MPI通信の高速化を実現した。

研究成果の学術的意義や社会的意義
ネットワークは制御できない静的な資源であるという前提から、動的に制御できる資源であるという視点に基づき、クラスタシステムのインターコネクト（相互結合網)を動的に制御可能なネットワーク資源ととらえることにより、近年の高性能計算機システム上での並列分散計算でのデファクトスタンダードなmessage passing方式に基づくMPIプログラムにより発生するパケットフローを動的に制御する。このことにより、新たな並列分散計算基盤の可能性を示した。「

研究成果の概要（英文）：In this research, acceleration of MPI communication was achieved based on a novel approach of seamlessly coordinating MPI program characteristics and network programmability brought by SDN. By overturning the assumption that network is a static resource that cannot be controlled, we dynamically control packet flows on the interconnect in a dynamic programming manner

研究分野：高性能計算

キーワード：MPI　SDN　OpenFlow

様　式　Ｃ－１９、Ｆ－１９－１、Ｚ－１９（共通）

１．研究開始当初の背景

近年、並列分散プログラミングライブラリ MPI(Message Passing Interface)の必要性と重要性が急激に高まりつつある。MPI は、MPI_Send、MPI_Recv などの１対１通信、MPI_Bcast、MPI_Reduce などの複数プロセスが関与する集合通信に関する API (Application Programming Interface) を提供し、開発者が高性能計算を行う計算機システムのネットワーク構成や特性を意識することなく、複数のプロセス間でメッセージを複雑に交換する分散並列プログラムを比較的容易に開発可能とする。また、今日の高性能計算機システムのアーキテクチャは、マルチコアを有する計算ノード群を高速ネットワークで結合したクラスタシステムへと急速に変遷しつつあり、その計算ノード数もまた年々増加傾向にある。MPI のプログラミングの容易性と、近年の高性能計算機システムのクラスタシステムへのアーキテクチャ変化によって、複数のノード上に分散するプロセッサコアを利用するための分散メモリプログラミングを可能にする MPI は、今日の計算機システム上で高性能計算を行う上でますます重要かつ必要不可欠な役割を担っている。一方で、大規模クラスタシステム上で MPI プログラムを高速に実行できるかどうかは、プログラム内で実行される MPI 通信、特に、複数のプロセスが関与する集合通信に要する時間をどれほど短縮できるかに大きく依存する。このような観点から、国内外においても MPI の集合通信の効率化を試みる研究開発が数多く報告されている。わが国のフラッグシップマシン京のインターコネクト Tofu 上で効率的に MPI_Allreduce を実現することを目的とした研究[1]、クラスタシステムのインターコネクト技術として利用される InfiniBand のハードウェアマルチキャスト機能を利用した MPI_Bcast の実現を目的とした研究などはその一例としてあげることができる。しかし、これらの先行研究は、クラスタシステムのインターコネクトは MPI プログラムから制御できない静的なネットワーク資源である、という前提に基づいていた。

２．研究の目的

本研究では、ネットワークは制御できない静的な資源であるという前提から、動的に制御できる資源であるという視点に基づき、クラスタシステムのインターコネクト (相互結合網)を動的に制御可能なネットワーク資源ととらえ、MPI プログラム特性と SDN のネットワークプログラミング性をシームレスに連携させるという新しいアプローチで、MPI 通信の高速化を実現することを目指す。具体的には、本研究では、インターコネクトとして OpenFlow 網を想定し、MPI プログラム実行時に発生する MPI 通信系列およびそれらの特性、インターコネクトトポロジ、利用状況、MPI プロセス配置情報を基に、MPI 通信によって発生するネットワークフロー系列を OpenFlow コントローラでプログラム制御することにより、MPI 通信ひいては MPI プログラムの高速実行を可能とする並列分散計算基盤を実現することを目的とする。

３．研究の方法

本研究目的達成のために、本研究では、

(1) MPI 通信エンコード機能の設計・実装
　インターコネクト上で発生するノード間のプロセス間通信種別をパケットにエンコードして埋め込む技術・技法を実現する。
(2) MPI 通信連携機能の設計・実装
　MPI 計算と連動して、ネットワーク制御を動的に制御する仕組み・フレームワークを実現する。
(3) 実環境での性能評価と有用性検証
　実際に並列分散計算基盤を実現し、当該計算基盤上において性能評価を行い、提案手法・技術の有用性を検証する。

の 3 点をマイルストーン課題とし、研究に取り組んだ。

４．研究成果

図 1 に本研究で提案・実装した MPI 通信連携機能の概要を示す。提案の基本的な考え方は、MPI が送出する各パケットに通信パターンをエンコードしたタグを付与し、タグに基づいてネットワークでパケット制御することにある。各計算ノードは、ユーザ空間で１つ以上の MPI アプリケーションのプロセスそ実行する。MPI アプリケーションは、MPI ライブラリを動的にリンクしている。計算ノードのカーネル空間には、「タグ付けカーネルモジョール」が常駐し、MPI ライブラリが送出するパケットに、タグ付け処理を実行する。MPI ライブラリとカーネルモジュールは連携動作し、タグ付けに必要な MPI ライブラリ内部の情報を共有する。

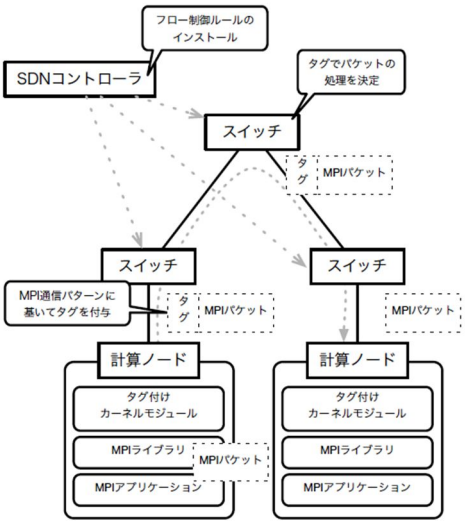

図 1: MPI 通信連携機能の概要.

SDN コントローラは、MPI パケットのタグに基づいて、パケットを制御するフローエントリを各スイッチのフローテーブルに登録する。スイッチは MPI パケットを受信するとパケットのタグを読み取り、フローテーブルにしたがってパケットを処理する。受信側の計算ノードに隣接するスイッチは、計算ノードにパケットを転送する前に、タグを除去する。

タグ付けカーネルモジュールは、MPI というアプリケーションのレイヤの情報を SDN で取り扱えるように、タグという形でパケットに埋め込む役割を担う。パケット自身に MPI 通信パターンがエンコードされ書き込まれているため、SDN コントローラやスイッチは、MPI アプリケーションと別途通信することなく、各々のパケットに対する制御を決定できる。本提案の MPI 通信連携機能により、低オーバヘッドな MPI アプリケーションとネットワーク制御の同期が可能となる。

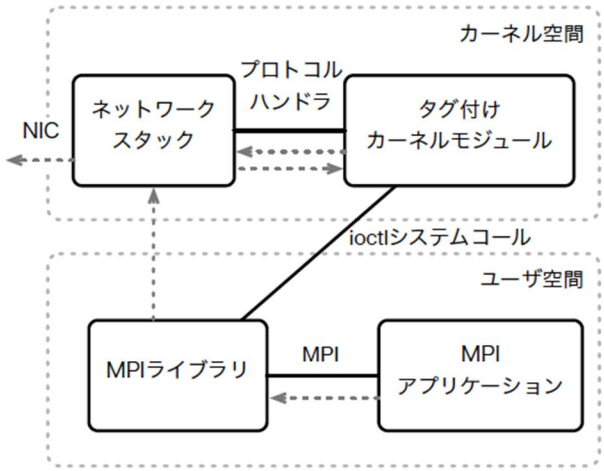

図 2: タグ付けカーネルモジュールと MPI ライブラリの連係.

図 2 に提案である MPI 通信連携機能を構成する構成要素の相互作用を示す。ユーザ空間のプログラムとカーネルモジュールの間の通信には profcs、sysfs、netlink ソケット、ioctl システムコールなど様々な実現方法がある。本研究では、転送するデータサイズが小さいことと、実装の容易さを考慮し、ユーザプログラムからデバイスドライバを制御するためのシステムコール ioctl を採用した。具体的な連携手順は以下の通りとなる。タグ付けカーネルモジュールは misc キャラクタデバイスとして振る舞い、/dev/sdnmpi にデバイスファイルを公開する。MPI ライブラリは、初期化時に/dev/sdnmpi デバイスを開き、ファイルハンドルを保持しておく。そして、MPI プロセス間で TCP 接続が確立した際(connect あるいは accept 関数成功時)に、/dev/sdnmpi デバイスに対して ioctl システムコールを発行し、ピア情報を引数として渡す。カーネルモジュール側では、ioctl システムコールのハンドラでピア情報を受信し、ピアリストに追加する。これにより、タグ付けカーネルモジュールが MPI プロセス間で TCP 接続が

確立したことを検知できる。

提案手法の有効性を評価するために、タグ付けによるオーバヘッドの大きさを評価する実験を行なった。マイクロベンチマークを利用して、2 ノード間での 1 対 1 通信の有効帯域幅と遅延を、提案手法を使用する場合としない場合で計測し比較した。評価には SDN スイッチに接続した 2 台の計算ノードを使用した。表 1 に SDN スイッチの性能を、表 2 に計算ノードの性能を示す。

表 1: SDN スイッチ.

型番	NEC ® UNIVERGE PF5240
SDN 規格	OpenFlow 1.0
Ports	Gigabit Ethernet ×48
スイッチング容量	176Gbps
転送性能	131Mpps

表 2: 計算ノード.

CPU	Intel ® Xeon ® CPU E5520 × 2
メモリ	12GB (ECC)
ネットワーク	Gigabit Ethernet
OS	Fedora 20 (Linux Kernel 3.19.8-100)

有効帯域幅の計測には OSU Micro Benchmark に含まれる osu_bw ベンチマークを使用し、ウインドウサイズは 64、繰り返し回数は 500 回とした。一方、遅延の計測には osu_latency を使用し、繰り返し回数は 50,000 回とした。有効帯域幅と遅延の計測の両方で、送信するメッセージサイズを 1 バイトから 2M バイトまで変化させた。

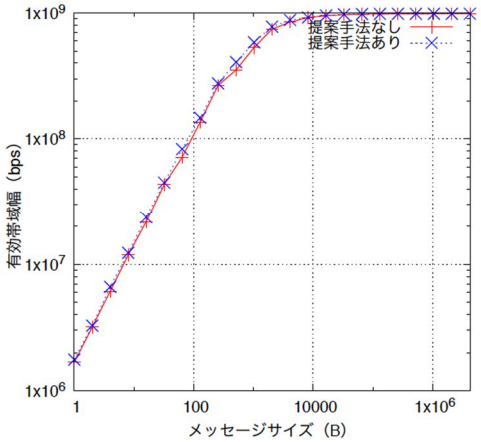

図 3: 2 ノード間の有効帯域幅の比較.

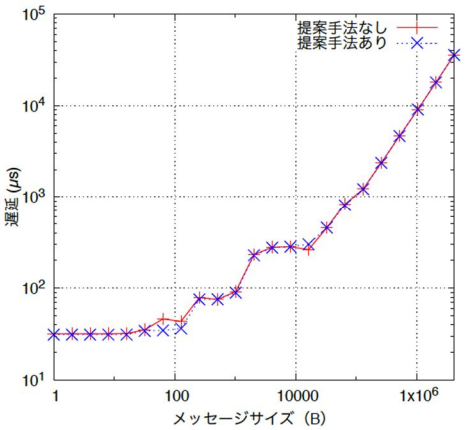

図 4:2 ノード間の遅延の比較.

図 3 および図 4 に有効帯域幅のベンチマーク結果、遅延のベンチマーク結果を示す。これらの結果において、全てのメッセージサイズで提案手法を使用する場合としない場合で大きな差は見られない。すなわち、提案する MPI 通信連携機能のタグ付けによるオーバヘッドは実用上無視でいる程度であるといえる。

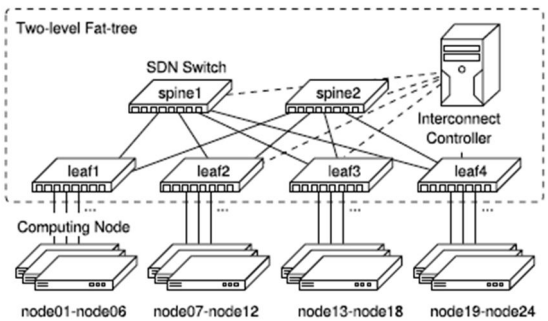

図 5:実環境での評価環境.

次に、実環境における MPI 通信連携機能の評価を行う。図 5 に評価環境を示す。SDN スイッチには、NEC ProgrammableFlow PF5240 を採用し、計算ノードには表 3 に示す SGI Rackable Half-Depth Server C1001 を用いた。本評価では MPI アプリケーションの実行に同期して相互結合網内を流れるパケットフローが動的に制御されているかどうかを検証する。具体的には、図 6 に示す MPI プログラムを計算ノード上で動作させ、MPI_Bcast のパケットフローを leaf1-spine1-leaf2 を通る経路に、MPI_Reduce のパケットフローを leaf1-spine2-leaf2 を通過するように制御する。この際の spine1 および spine2 でスループット変化を観測し、MPI 実行に連動した動的経路制御が実現できているかどうかを検証する。

表 3: 計算ノード.

Name	Spec
CPU	Intel Xeon E5-2620 2.00GHz 6core × 2
Memory	64GB (DDR3-1600 8GB × 8)
Network	Gigabit Ethernet
OS	CentOS 7.2
Kernel	Linux 3.10
MPI Library	MPICH 3.1.4

```
#include <mpi.h>
#define BUF_SIZE (1000)
#define REPEAT_COUNT (10000)

char send_buf[BUF_SIZE];
char recv_buf[BUF_SIZE];

int main(int argc , char** argv) {
    MPI_Init(&argc , &argv);

    /* Record current time as t1 */

    /* MPI_Bcast */
    for (i = 0; i < REPEAT_COUNT; i++) {
        MPI_Bcast(send_buf, BUF_SIZE, MPI_CHAR, 0,
                  MPI_COMM_WORLD);
    }

    /* Record current time as t2 */

    /* MPI_Reduce */
    for (i = 0; i < REPEAT_COUNT; i++) {
        MPI_Reduce(send_buf, recv_buf, BUF_SIZE,
                   MPI_CHAR, MPI_SUM, 0,
                   MPI_COMM_WORLD);
    }

    /* Record current time as t3 */

    MPI_Finalize();
}
```

図 6: 評価に用いる MPI アプリケーション.

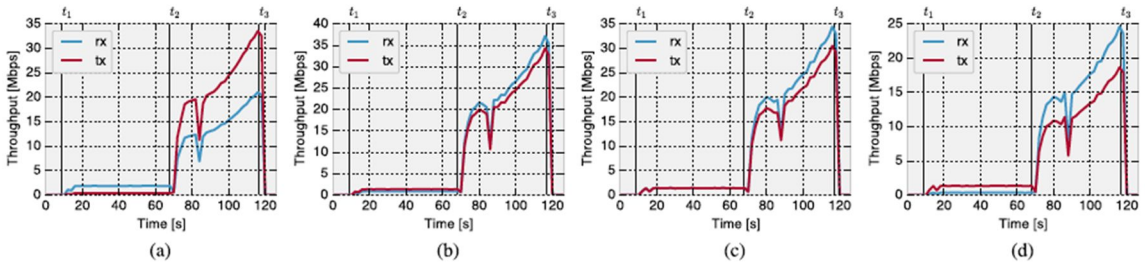

図 7: spine1 のポートで計測されるスループット.

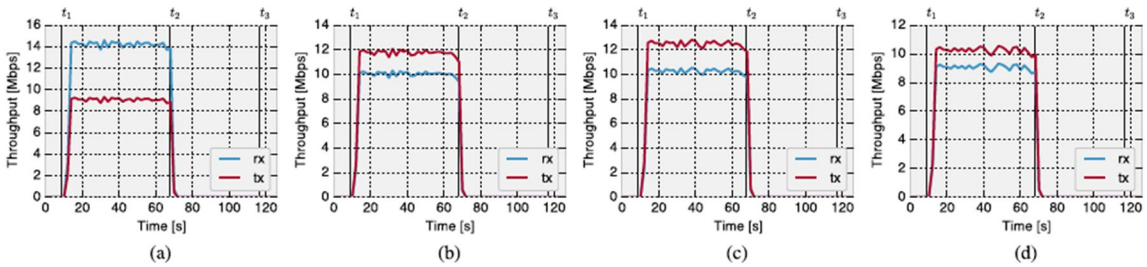

図 8: spine2 のポートで計測されるスループット.

図 7 および図 8 に spine1 および spine2 で計測されたスループットを示す。この結果より、MPI_Bcast が実行され MPI__Reduce が実行されたタイミング、すなわち、MPI アプリケーションの実行にあわせて経路制御が実現できていることがわかる。

以上の結果から、本研究では、ネットワークは制御できない静的な資源であるという前提から、動的に制御できる資源であるという視点に基づき、クラスタシステムのインターコネクト (相互結合網)を動的に制御可能なネットワーク資源ととらえ、MPI プログラム特性と SDN のネットワークプログラミング性をシームレスに連携させるという新しいアプローチで、MPI 通信の高速化を実現することができた。

５．主な発表論文等

〔雑誌論文〕　計4件（うち査読付論文　2件／うち国際共著　0件／うちオープンアクセス　2件）

１．著者名 Endo Arata、Ohtsuji Hiroki、Hayashi Erika、Yoshida Eiji、Lee Chunghan、Date Susumu、Shimojo Shinji	４．巻 8
２．論文標題 Dynamic Traffic Control of Staging Traffic on the Interconnect of the HPC Cluster System	５．発行年 2020年
３．雑誌名 IEEE Access	６．最初と最後の頁 198518～198531
掲載論文のDOI（デジタルオブジェクト識別子） 10.1109/ACCESS.2020.3035158	査読の有無 有
オープンアクセス オープンアクセスとしている（また、その予定である）	国際共著 －

１．著者名 Keichi Takahashi, Susumu Date, Dashdavaa Khureltulga, Yoshiyuki Kido, Hiroaki Yamanaka, Eiji Kawai, Shinji Shimojo	４．巻 6
２．論文標題 UnisonFlow: A Software-Defined Coordination Mechanism for Message-Passing Communication and Computation	５．発行年 2018年
３．雑誌名 IEEE Access	６．最初と最後の頁 23372-23382
掲載論文のDOI（デジタルオブジェクト識別子） 10.1109/ACCESS.2018.2829532	査読の有無 有
オープンアクセス オープンアクセスとしている（また、その予定である）	国際共著 －

１．著者名 Date Susumu、Yoshikawa Takashi、Nozaki Kazunori、Watashiba Yasuhiro、Kido Yoshiyuki、Takahashi Masahiko、Muraki Masaya、Shimojo Shinji	４．巻 1
２．論文標題 Towards A Software Defined Secure Data Staging Mechanism	５．発行年 2017年
３．雑誌名 Sustained Simulation Performance 2017	６．最初と最後の頁 15～24
掲載論文のDOI（デジタルオブジェクト識別子） 10.1007/978-3-319-66896-3_2	査読の有無 無
オープンアクセス オープンアクセスではない、又はオープンアクセスが困難	国際共著 －

１．著者名 Takahashi Keichi、Date Susumu、Watashiba Yasuhiro、Kido Yoshiyuki、Shimojo Shinji	４．巻 1
２．論文標題 Integrating SDN-Enhanced MPI with Job Scheduler to Support Shared Clusters	５．発行年 2020年
３．雑誌名 Sustained Simulation Performance 2018 and 2019	６．最初と最後の頁 149～159
掲載論文のDOI（デジタルオブジェクト識別子） 10.1007/978-3-030-39181-2_13	査読の有無 無
オープンアクセス オープンアクセスではない、又はオープンアクセスが困難	国際共著 －

〔学会発表〕　計7件（うち招待講演　0件／うち国際学会　5件）

1．発表者名
Shinji Shimojo, Susumu Date,

2．発表標題
Osaka University's Research Infrastructure Towards Acceleration of Global e-Science Research

3．学会等名
17th IEEE International Conference on eScience 2021（国際学会）

4．発表年
2021年

1．発表者名
遠藤 新, Chunghan Lee, 伊達 進, 木戸 善之, 渡場 康弘, 下條 真司

2．発表標題
ステージングによるトラヒック競合を自動抽出可能なパケットフロー分析ツール

3．学会等名
第17回 ディペンダブルシステムワークショップ, 日本ソフトウェア科学会

4．発表年
2019年

1．発表者名
Keichi Takahashi, Susumu Date, Yasuhiro Watashiba, Yoshiyuki Kido, Shinji Shimojo

2．発表標題
Integrating SDN-Enhanced MPI with Job Scheduler to Support Shared Clusters

3．学会等名
28th Workshop on Sustained Simulation Performance（国際学会）

4．発表年
2019年

1．発表者名
Yohei Takigawa, Keichi Takahashi, Susumu Date, Yoshiyuki Kido, Shinji Shimojo

2．発表標題
A Traffic Simulator with Intra-node Parallelism for Designing High-performance Interconnects

3．学会等名
The 2018 International Conference on High Performance Computing & Simulation (HPCS 2018)（国際学会）

4．発表年
2018年

１．発表者名
遠藤新，伊達進，木戸善之，渡場康弘，下條真司

２．発表標題
インターコネクトにおけるステージング I/O 通信とノード間通信の分離による性能評価

３．学会等名
日本ソフトウェア科学会 第16回ディペンダブルシステムワークショップ

４．発表年
2018年

１．発表者名
Keichi Takahashi, Susumu Date, Khureltulga Dashdavaa, Yoshiyuki Kido, Shinji Shimojo

２．発表標題
PFAnalyzer: A Toolset for Analyzing Application-aware Dynamic Interconnects

３．学会等名
The Monitoring and Analysis for High Performance Computing Systems Plus Applications (HPCMASPA) Workshop, Cluster 2017 (国際学会)

４．発表年
2017年

１．発表者名
Hiroaki Morimoto, Khureltulga Dashdavaa, Keichi Takahashi, Yoshiyuki Kido, Susumu Date, Shinji Shimojo

２．発表標題
Design and Implementation of SDN-enhanced MPI Broadcast Targeting a Fat-tree Interconnect

３．学会等名
The 2017 International Conference on High Performance Computing & Simulation (HPCS 2017) (国際学会)

４．発表年
2017年

〔図書〕　計0件

〔産業財産権〕

〔その他〕

―

６．研究組織

	氏名 （ローマ字氏名） （研究者番号）	所属研究機関・部局・職 （機関番号）	備考
研究協力者	高橋　慧智 (Takahashi Keichi) (40846408)	東北大学・サイバーサイエンスセンター・助教 (11301)	

６．研究組織（つづき）

	氏名 （ローマ字氏名） （研究者番号）	所属研究機関・部局・職 （機関番号）	備考
研究協力者	遠藤　新 (Endo Arata) (20895271)	奈良先端科学技術大学院大学・総合情報基盤センター・助教 (14603)	

７．科研費を使用して開催した国際研究集会

〔国際研究集会〕　計0件

８．本研究に関連して実施した国際共同研究の実施状況

共同研究相手国	相手方研究機関