ポストペタスケールシステム向けの並列計算モデルの開発と評価

課題責任者
上原　均　　海洋研究開発機構　地球情報基盤センター

著者
上原　均　　海洋研究開発機構　地球情報基盤センター
横川　三津夫　　神戸大学　大学院システム情報学研究科
村井　均　　理化学研究所　計算科学研究機構
板倉　憲一　　海洋研究開発機構　地球情報基盤センター
浅野　俊幸　　海洋研究開発機構　地球情報基盤センター

3 ～ 5 年後と予測されるポストペタスケール級計算機の時代では 100 万並列にまで対応可能な高並列プログラムが必要となるが、そのような高並列プログラムの開発は困難である事も指摘されている。特に地球科学分野のプログラムは時に数十万行以上に及ぶことから事前の移行検討が不可欠である。それらをふまえると、既存の計算機環境でも利用でき、かつポストペタ時代にも通用する並列プログラム開発手段があれば極めて有効といえる。そこで我々は、ポストペタ時代において有望と目される PGAS モデルに着目し、それに基づいた XcalableMP 言語を、多くの地球科学分野のプログラムが稼働する地球シミュレータ上で評価した。今年度の本研究では、特に高並列計算において性能に顕著な影響を与えうる通信性能を中心に性能評価を行った。計算科学系計算プログラムで多く見られる袖領域通信パターンなどについて、XcalableMP 言語で記述されたプログラムと、現行のユーザが多く用いている Message Passing Interface で記述した際の通信性能を測定し、比較検討を行った。

キーワード：ポストペタ, 並列プログラミング言語, PGAS, 性能評価

1. はじめに

近年、計算機ユーザからの計算性能向上に対する強い要望に応えて高性能計算機の性能は著しく向上しており、計算機システムの総 Core 数は数万から数十万に達している。さらに 3 ～ 5 年後に登場すると予測されるポストペタスケール級計算機の総 Core 数は 100 万に達する事が見込まれる。一方で、そのような高並列計算機を活用するために不可欠な高並列プログラムの開発は、開発に要する人的コストの高騰や技術的難易度の上昇、開発期間の長期化などから、現在の並列プログラム開発よりも遥かに難しくなる事が専門家らによって予測されている。特に地球科学分野のプログラムは、時にその行数が数十万行以上に及ぶことから、コストが更に膨大なものになることが推測できるため、早期の移行対策が重要といえる。この早期の移行対策には既存の計算機環境でも利用でき、かつポストペタスケール時代に通用する並列プログラム開発手段があれば極めて有効である。

高並列プログラム開発に資する次世代並列モデルの研究は世界的に進められており、なかでも区分化大域アドレス空間（Partitioned Global Address Space、以下 PGAS と略記）[1][2] モデルは普及が見込まれており、Fortran2008 の言語規格にも導入され、ECMWF などの地球科学の研究機関でも検討が進められている。

それらの背景をふまえて本研究では、ポストペタスケール級計算機における高並列プログラム開発に資するための並列計算モデル（並列プログラミング言語）として、理化学研究所が開発している PGAS 言語 XcalableMP[3] に注目し、地球シミュレータへの移植および性能評価・分析などを通じて、その利用可能性を検討する。XcalableMP 言語はその生産性の高さが国際的にも評価されており [4]、QCD シミュレーションなどでの評価も進められているが、一方で地球シミュレータのようなベクトル型計算機での実績や地球科学分野での評価事例は比較的乏しい。今年度は、高並列・大規模シミュレーションの実施に備えて、高並列時において性能に大きな影響を与える要因の一つである通信性能に焦点を当てて検討を行った。

なお本研究は海洋研究開発機構と神戸大学、理化学研究所による共同研究「ポストペタスケールシステム向けの並列計算モデルの開発と評価」の一環として実施した。

2. 研究計画

本研究は以下の段階に大別して実施した。

1) XcalableMP コンパイラ [5] の地球シミュレータへの移植
2) 地球シミュレータでの XcalableMP 記述プログラムの性能評価
   a) 基礎的評価としての、袖通信および Broadcast 通信での性能評価
   b) 実用的性能評価としての、NICAM-DC-mini[6][7] での性能評価

1) の XcalableMP コンパイラの移植については Stable version 1.1.0[5] を移植した。当該バージョンは地球シミュレータを構成する NEC SX-ACE に対応しているため、移植上の問題は発生しなかった。2) の性能評価は次章で述べる。

3. 今年度の成果

本章では地球シミュレータ上での XcalableMP 言語の通信性能評価を述べる。この評価では、先述の通り、実アプリケーションにしばしばみられる袖通信や Broadcast 通信を用いた基礎的性能評価、と実用的性能評価を意図した NICAM-DC-mini での評価を行った。

3.1 基礎的性能評価

袖通信と Broadcast 通信による基礎的性能評価では、XcalableMP 言語の持つグローバルビューモデルとローカルビューモデルの性能差も検討した。XcalableMP 言語のグローバルビューモデルとは各ノードに共有して利用するデータを各ノードに分散配置する方式であり、ローカルビューモデルは各ノードが持つローカルデータに対して通信を行う方式である。従来の通信ライブラリである MPI で実装した袖通信プログラムおよび Broadcast 通信プログラムをそれぞれグローバルビューモデルとローカルビューモデルに書換えて、三者の性能比較を行った。

3.1.1 袖通信プログラムでの評価

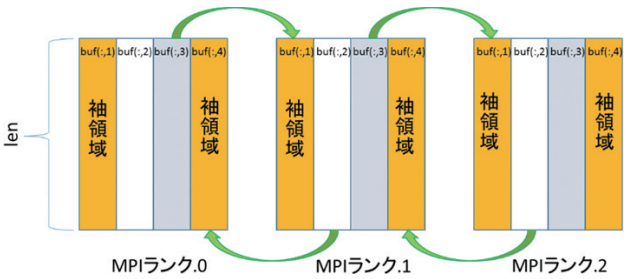


図 1　袖通信プログラムの通信イメージ

図 1 に示すような通信を行う袖通信プログラムを XcalableMP 言語のグローバルビューとローカルビューで書換えた。グローバルビューでの書換えでは、データ分割を指示行で書き、MPI_Send 関数などで書かれた通信部分を XcalableMP 言語の reflect 構文で記述した。このプログラムでは通信データ長が動的に変化するが、reflect 構文では配列全体が通信対象となるため MPI 記述時のように通信データ個数を限定する事が出来ない。そのため、通信区間をサブルーチン化することで対応した。ローカルビューへの書換えは MPI_Send 関数などで書かれた部分を Co-array Fortran（CAF）の片側通信に書き換えた。これら MPI プログラム（MPI）とグローバルビュー形式（XMP-G）、ローカルビュー形式（XMP-L）の性能測定結果を図 2、図 3 に示す。図は横軸がメッセージ長であり、縦軸に 100 回試行した際の平均所要時間を示している。短メッセージ時は MPI 記述がやや高速であったが、長メッセージ時ではグローバルビュー形式が高速であった。これはグローバルビューの内部実装に MPI_Send_init 等の高速な通信関数が用いられているためと考えられる。ローカルビュー形式は性能面で劣るものの、簡潔な通信記述が可能であることを確認した。

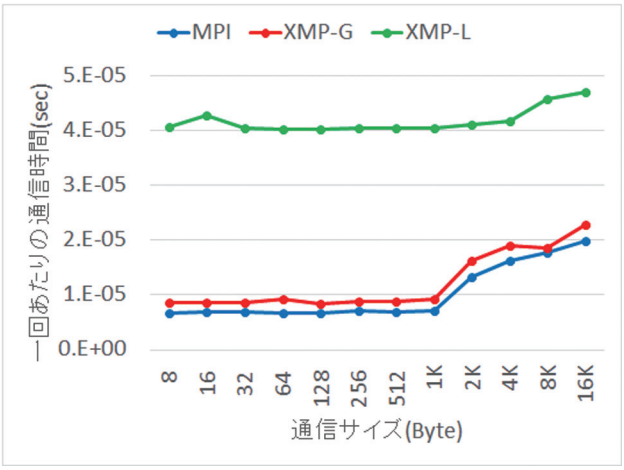


図 2　袖通信（短メッセージ長時）

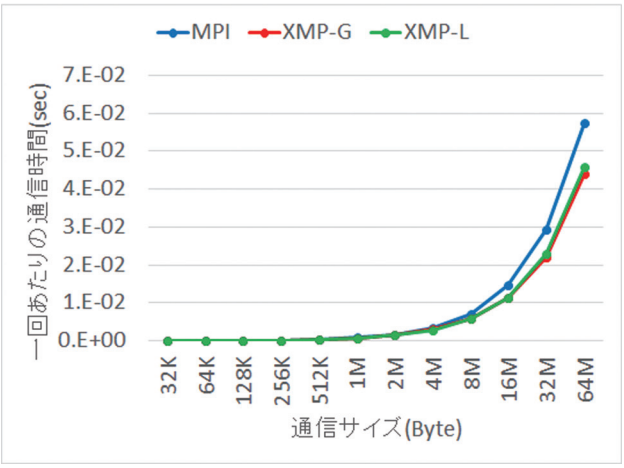


図 3　袖通信（長メッセージ長時）

3.1.2 Broadcast 通信プログラムでの評価

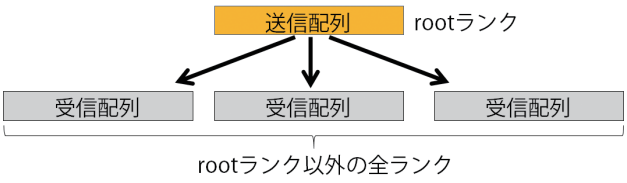


図 4　Broadcast 通信のイメージ

図 4 に示されるような Broadcast 通信の MPI プログラムを XcalableMP 言語のグローバルビュー形式およびローカルビュー形式で書換えた。グローバルビュー形式では BCAST 指示文を、ローカルビュー形式では CAF の CO_BROADCAST 関数を用いた。どちらも指定配列の全要素が通信対象となるため、袖通信プログラムのグローバルビュー形式同様のサブルーチン化で対応した。図 5 ～ 8 に計測結果を示す。横軸はノード数、縦軸は 100 回試行時の平均所要時間を示す。この計測結果から XcalableMP 言語実装が従来の MPI 記述とほぼ同等な性能を示す事を確認できた。

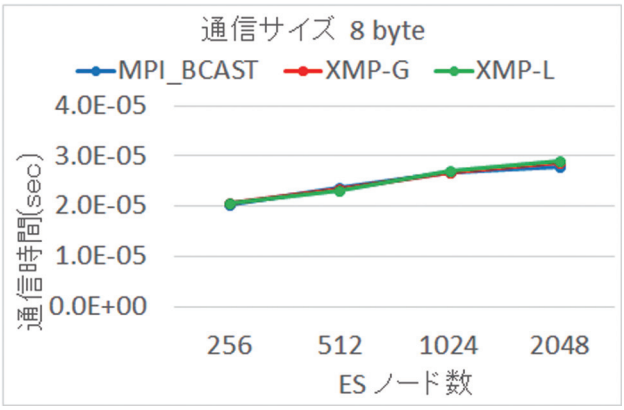


図 5　8 バイト長での Broadcast 通信比較

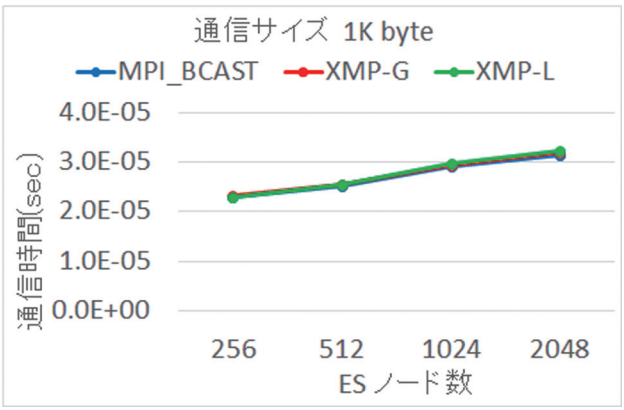


図 6　1K バイト長での Broadcast 通信比較

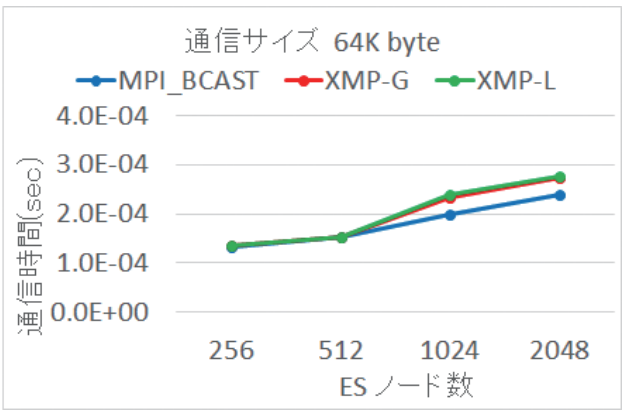


図 7　64KB 長での Broadcast 通信比較

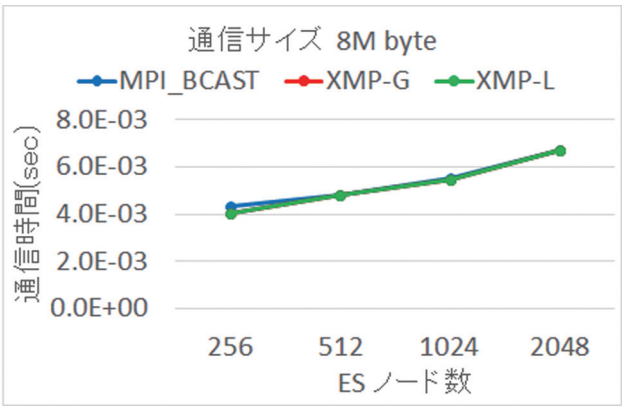


図 8　8MB 長での Broadcast 通信比較

3.2　NICAM-DC-mini での実用的性能評価

より実アプリケーションに近い形での実用的性能評価としてミニアプリ集 Fiber[6] にふくまれる NICAM-DC-mini について、MPI 版 [6] と XcalableMP 言語版 [7] で比較を行った。

XcalableMP 言語版はローカルビュー形式である。評価には jablonowski（GLevel05，RLevel00，40 層，10 並列）を用いた。NICAM-DC-mini の通信は 1）隣接間 1 対 1 通信、2）ブロードキャスト通信、3）リダクション演算を伴う集団通信、に大別される。それぞれの計測結果を図 9 ～ 11 に示す。XcalableMP 言語による実装では全体的に通信時間の増大が確認された。また MPI 実装時と比べて演算負荷の変化も見受けられた。

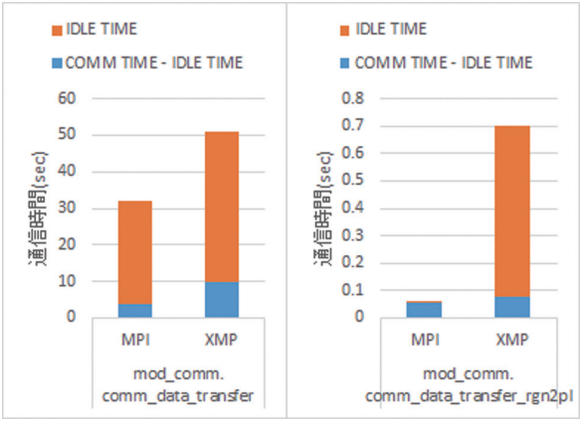


図 9　隣接間 1 対 1 通信

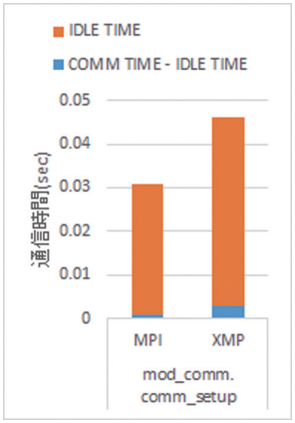


図 10　ブロードキャスト通信

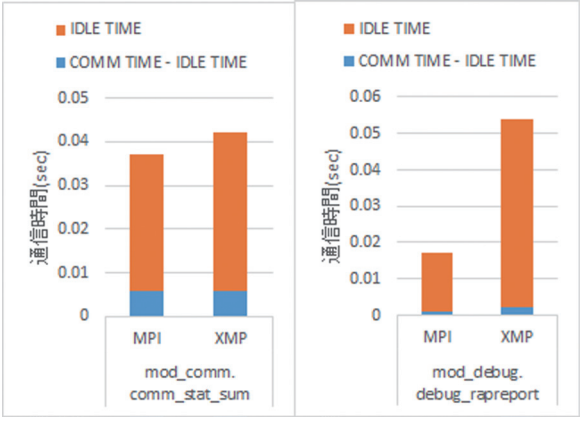


図 11　リダクション演算を伴う集団通信

4. おわりに

本研究では PGAS 言語 XcalableMP の地球シミュレータでの評価を行った。今年度は特に通信性能を中心とした性能評価を行った。袖通信、Broadcast 通信による基礎的な通信性能評価では従来の MPI プログラムより良好な性能が一部確認された。しかし NICAM-DC-mini を用いた実用的性能評価では、XcalableMP 言語による実装では全体的に通信時間の増大が確認された。その詳細原因の検討および改善が今後の課題である。また、より高度な実アプリケーションや数値計算プログラム以外の、例えば可視化プログラムなどへの適用も今後の課題といえる。

謝辞

本研究の一部は文部科学省フラッグシップ 2020 プロジェクト（ポスト「京」の開発）「ポスト「京」で重点的に取り組むべき社会的・科学的課題」における重点課題④「観測ビッグデータを活用した気象と地球環境予測の高度化」（課題責任者：海洋研究開発機構 高橋桂子）の一環として行われた。また本研究の実施においては、NEC ソリューションイノベータ株式会社 山口健太氏ほか、日本電気株式会社関係各位に様々なご協力をいただいたことに感謝し、ここに記す。

文献

[1] Bill Carlson et al., "Programming in the Partitioned Global Address Space Model", http://upc.gwu.edu/tutorials/tutorials_sc2003.pdf
[2] Partitioned Global Address Space, http://www.pgas.org/
[3] XcalableMP, http://www.xcalablemp.org/
[4] Masahiro Nakao et al., “XcalableMP for Productivity and Performance in HPC Challenge Award Competition Class 2”, HPCC Class2 Submission, Denver, Colorado, USA, Nov. 2013.
[5] Omni Compiler Project, http://omni-compiler.org/
[6] Fiber, http://fiber-miniapp.github.io/
[7] 村井均、PGAS 言語 XcalableMP による Fiber ミニアプリ集の実装と評価、2016 年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 (HPCS2016), 2016.

Study of Parallel Computational Model for Post-Peta Scale Computer System

Project Representative

Hitoshi Uehara — Center for Earth Information Science and Technology, Japan Agency for Marine-Earth Science and Technology

Authors

Hitoshi Uehara — Center for Earth Information Science and Technology, Japan Agency for Marine-Earth Science and Technology

Mitsuo Yokokawa — Graduate School of System Informatics, Kobe University

Hitoshi Murai — Advanced Institute for Computational Science, RIKEN

Ken'ichi Itakura — Center for Earth Information Science and Technology, Japan Agency for Marine-Earth Science and Technology

Toshiyuki Asano — Center for Earth Information Science and Technology, Japan Agency for Marine-Earth Science and Technology

Experts have inferred that a parallel programming language based on the partitioned global address space (PGAS) model could be a promising prospect for the development of large scale parallel programs that are required for hundreds of petaFLOPS class (post petascale) computers. The purpose of this study is to evaluate the applicability of XcalableMP, which is a type of PGAS-based language. In this study, we measured the communication performance of a benchmark coded in both Fortran 90 using Message Passing Interface (MPI) and XcalableMP on the Earth Simulator. Based on the benchmark results, we conclude that XcalableMP codes can achieve a better performance than original MPI codes in some cases.

Keywords: Post Peta scale, parallel programming language, PGAS model, Benchmark

The partitioned global address space (PGAS) model [1] is a promising parallel computational model for the forthcoming era involving hundreds of petaFLOPS class (post petascale) computers. In particular, XcalableMP [2], which is a PGAS-based language, exhibits the two desirable features of code readability and ease of rewriting from legacy Fortran/C code. Thus, XcalableMP is promising for the development of large-scale programs. It would be considerably useful in preparing for the forthcoming post petascale era if the XcalableMP language could be employed on existing computer environments. One representative environment in the field of earth science is the Earth Simulator. Many earth science programs are executed on the Earth Simulator. We previously constructed an environment on which the XcalableMP language can be employed on the Earth Simulator, and have confirmed the applicability of XcalableMP to the Earth Simulator [3]. However, so far we have mainly evaluated the computational performance of the XcalableMP code, and have not focused on the communication performance.

The communication performance would be one of the important performance factors in realizing large scale parallel executions. Therefore, we aim to evaluate the communication performance of XcalableMP codes on the Earth Simulator. We have employed the Omni XcalableMP Compiler stable version 1.1.0 [4] as the XcalableMP compiler on the Earth Simulator for the following benchmarks.

First, we have measured both the shift communication code and broadcast communication code, as basic performance benchmarks. These are coded with Fortran 90 using Message Passing Interface (MPI) and XcalableMP. Regarding the XcalableMP implementation, either local-view or global-view methods can be employed. In the global-view implementation, multiple processes share a global array, distributed using directive statements. In the local-view implementation, each process owns a local array, and these processes communicate using local-arrays, similarly to MPI programing. For the basic benchmarks, we have implemented each benchmark in both the global-view and local-view styles, and have compared these with the original Fortran 90 (MPI) implementation. We measured the elapsed time for 100 communication iterations.

Figure 1 presents the measurement results for shift communication. The vertical axis represents the average elapsed time per iteration, and the horizontal axis represents the message length. The XcalableMP global-view implementation achieves a better performance than the MPI implementation in the case of a long message length.

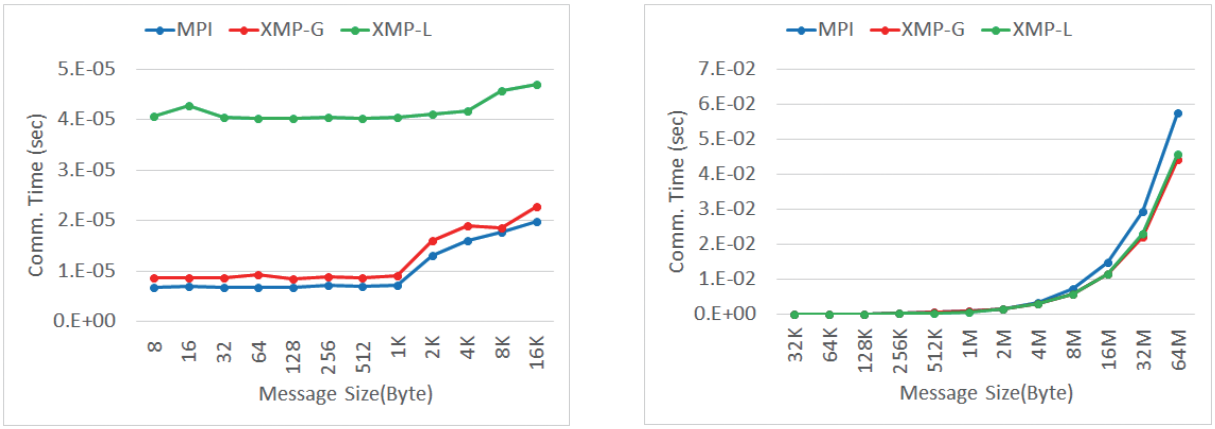
We present the measurement results for broadcast communication in Fig.2. The vertical axis represents the

average elapsed time per communication, and the horizontal axis represents the number of processor nodes used. In the measurement, only one process is assigned to each node. Thus, the number of processor nodes used is equal to the number of processes. Figure 2 shows that the performance of the XcalableMP code is as about the same as those achieved by the MPI codes.

Second, as a more practical application we measured the communication performance of NICAM-DC-mini. NICAM-DC [5] is the dynamical core of NICAM (the Nonhydrostatic Icosahedral Atmospheric Model). NICAM-DC-mini is the kernel code of NICA-DC, and is included in the mini-application collection FIBER [6]. Murai has implemented NICAM-DC-mini in XcalableMP [7]. The XcalableMP version is implemented in local-view. We measured the performance of both the original NICAM-DC-mini coded using MPI and the XcalableMP version on the Earth Simulator. To perform the measurement, we employed the data set named "jablonowski" (GLevel05, RLevel00, 40 layers), and executed this using 10 processes. The communications of NICAM-DC-mini are classified into three types: (a) peer-to-peer communication, (b) broadcast communication, and (c) reduction communication. The measurement results are presented in Fig.3.

As seen in Fig. 3, we have confirmed an increase in communication times using the XcalableMP code. We infer that the major reason for this is the overhead in the XcalableMP implementation.

To summarize this study, we have benchmarked the



(a) Short message case (b) Long message case

Fig. 1 Measurement result of shift communication.

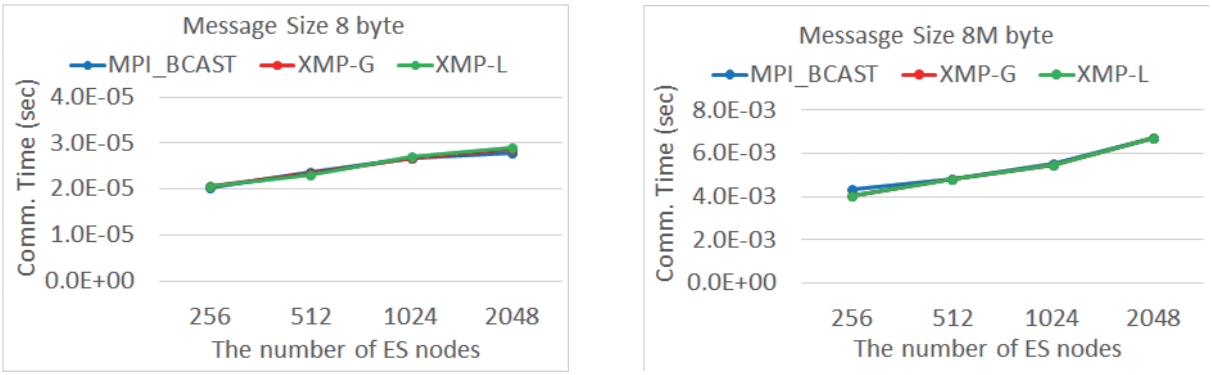
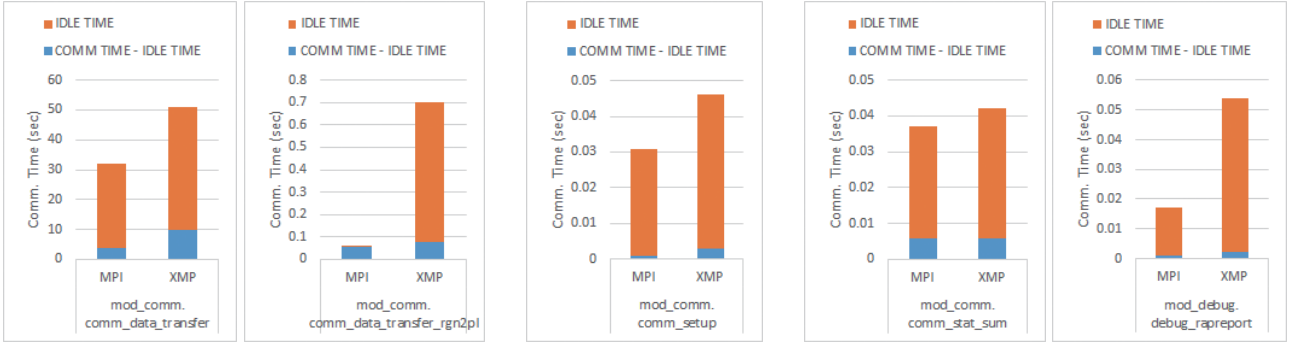


Fig. 2 Measurement result of broadcast communication.



(a) Peer-to-peer comm. (b) Broadcast comm. (c) Reduction comm.

Fig. 3 Measurement results of the communication part in NICAM-DC-mini.

communication performances of XcalableMP codes on the Earth Simulator. We have confirmed that XcalableMP codes can sometimes achieve better performances than the original MPI codes. However, the XcalableMP implementation still requires improvements in its performance in certain aspects. In future work, we plan to attempt to realize more realistic applications or visualization codes in XcalableMP.

Acknowledgement

A part of this study was conducted as part of post-K priority issue 4 “Advancement of meteorological and global environmental predictions utilizing observational Big Data.” This work was supported by Mr. Kenta YAMAGUCHI and other NEC staffs, who are invaluable to the completion of this study. We are deeply grateful to them.

References

[1] Partitioned Global Address Space, http://www.pgas.org/
[2] XcalableMP, http://www.xcalablemp.org/
[3] Hitoshi U., et al., “Study of Parallel Computational Model for Post-Peta Scale Computer System”, Annual Report of the Earth Simulator April 2015 - March 2016, 2016.
[4] Omni Compiler Project, http://omni-compiler.org/
[5] NICAM-DC, http://scale.aics.riken.jp/nicamdc/
[6] FIBER, http://fiber-miniapp.github.io/
[7] Hitoshi M., “Implementation and Evaluation of the Fiber miniapps using a PGAS language XcalableMP”, Organized session, HPCS2016 (2016).