Vol.2015-HPC-148 No.32
2015/3/3

Tofu インターコネクト 2 上での ACP 基本層の実装と性能評価

野瀬貴史[†1,†2]　安島雄一郎[†1,†2]　佐賀一繁[†1,†2]　志田直之[†1,†2]　住元真司[†1,†2]

Advanced Communication for Exa (ACE)プロジェクトにおいて開発している Advanced Communication Primitives (ACP) の Tofu インターコネクト実装を Tofu インターコネクト 2 (Tofu2) に移植し、その性能を評価した。Tofu2 はただデータ転送速度を向上するのみならず、ACE プロジェクトの過去の検討を踏まえた不可分操作機能が追加されている。ACP の Tofu 版では不可分操作はソフトウェア実装されていたが、Tofu2 版ではインターコネクトの機能を利用しハードウェア化した。ソフトウェア実装とハードウェア実装の性能を評価した結果，ハードウェア実装のリモートデータの操作がソフトウェア実装に比べ約 5.1 マイクロ秒改善した．一方，富士通 MPI の拡張インターフェースに実装されている不可分操作と比べると，約 7.5 マイクロ秒の開きがあった．これは現在の ACP の実装が，メインスレッドと通信側スレッドに分かれていることに起因し，レイテンシの削減に課題がある．

Implementation and Performance Evaluation of ACP Basic Layer on the Tofu Interconnect 2

TAKAFUMI NOSE[†1,†2]　YUICHIRO AJIMA[†1,†2]　KAZUSHIGE SAGA[†1,†2]
NAOYUKI SHIDA[†1,†2]　SHINJI SUMIMOTO[†1,†2]

We ported the Advanced Communication Primitives (ACP) library, that is developed under the Advanced Communication for Exa (ACE) project, for the Tofu interconnect (Tofu1) to the Tofu interconnect 2 (Tofu2). The Tofu2 not only improved the data transfer speed but also added a new feature, atomic operation, which is based on our past research. On the Tofu1, ACP's atomic operation functions were implemented by pure software. On the Tofu2, we modified ACP's atomic operation functions to utilize the hardware feature of the Tofu2. We evaluated the software implementation and the hardware implementation on the PRIMEHPC FX100. Remote data operation of the hardware implementation was about 5.1 microseconds faster than that of the software implementation. On the other hand, the hardware implementation was about 7.5 microseconds slower than the atomic operation function of the Fujitsu MPI Extended RDMA interface. This was caused by ACP's structure that is separated into a main thread and a communication thread. We will continue the development of ACP and reduce the latency in the future.

1. はじめに

ACE （Advanced Communication for Exa）プロジェクト[1]では，エクサスケールに向けて省メモリ・低遅延を両立する通信ライブラリ ACP (Advanced Communication Primitives)[2] の開発に取り組んでいる．既存の通信ライブラリは通信用バッファおよび管理制御用のバッファを暗黙に管理しており，通信の挙動やプロセス数に影響されてメモリ消費が激しくなるという問題がある．この問題はエクサスケールに向けてノード数が増加かつコアあたりのメモリ容量が減少すると顕著になるため，省メモリな通信ライブラリが必須である．ACP ライブラリでは，通信およびデータ配置をアプリケーションの側で明示的に制御することで付随するメモリ使用量の制御を可能とし，既存ライブラリの問題点を解決している．ACP にはデータ配置を最適化するための分散動的データ構造インターフェースが備わっており，この基礎にはグローバルメモリアロケータが位置する．分散データ構造の管理情報はローカルおよびリモートの複数のプロセスがアクセスするため，排他に制御される必要がある．ノード数が増加するとデータがますます分散され，リモート操作回数やアクセスの競合が増えると予想されるため，分散データ構造の排他制御の低遅延化や遅延の隠蔽は重要である．また，そのようなデータ構造はリモートアクセス削減のため局所性を持って配置されるため，排他制御性能の局所性も重要となる．我々は過去，グローバルデータ構造の効率的な操作, 非同期グローバルヒープ, およびグローバルメモリ管理モデルの検討を行い，インターコネクトに不可分操作があるとデータ構造の排他制御の性能が大きく改善できること[3]，さらに不可分操作においてプロセッサとインターコネクトが緊密に連携することでローカルからのアクセスレイテンシが大きく改善できることを示した[4][5]．また，将来そのようなハードウェアが登場することを前提として，ACP のメモリモデルと API を設計してきた．

今回，インターコネクトが持つべき機能に関する我々の上記見解が反映された，新しいハードウェアである Tofu インターコネクト 2（Tofu2）上に ACP を移植, および Tofu2 固有の機能を利用して不可分操作をハードウェア化し，その性能を評価したので報告する．

†1 富士通株式会社　次世代テクニカルコンピューティング開発本部
Fujitsu Limited., Next Generation Technical Computing Unit
†2 独立行政法人科学技術振興機構　戦略的創造研究推進事業
Japan Science and Technology Agency (JST),
Core Research for Evolutional Science and Technology (CREST)

2. ACP 基本層

ACP 基本層はグローバルメモリモデルを採用した通信ライブラリである. 基本層は ACP スタックの最も基礎に位置し，インターコネクトの機能を抽象化する役割を負っている[6]．各プロセスにあるメモリ領域が ACP ライブラリに登録されると, グローバルアドレスを取得することができ，このグローバルアドレスを用いることで全プロセスからこのメモリ領域にアクセスできる，というメモリモデルを ACP は採用している．グローバルアドレス長は 64bit である．ACP 基本層には 64bit のデータ長に対する不可分操作インターフェースがあり，これはグローバルアドレス自体を不可分操作の対象にすることを可能とするためのものである．ACP 基本層の不可分操作に関する機能は，上位の分散データ構造ライブラリやユーザーアプリケーションから活用される．データコピーのインターフェースは memcpy 関数に似たインターフェースを持つ acp_copy 関数で提供される．acp_copy 関数のデータのコピー元，及びコピー先の指定は，両方ともリモートプロセスのグローバルアドレスであっても良い．ローカルプロセスは，データのコピー元プロセスにデータコピーの実施を依頼するため，リモートプロセス間でのデータコピーはローカルプロセスを経由することがない．現在，基本層は UDP 版と Tofu 版が実装されている[7]が，どちらも不可分操作およびリモート間コピーはソフトウェア実装されている．

3. Tofu インターコネクト 2

Tofu2[8]は「京」および PRIMEHPC FX10 に使われている Tofu インターコネクト（以下区別のため Tofu1 と呼称する）[9]を継承し発展させたインターコネクトであり，リンクバンド幅が 5GB/s×2 から 12.5GB/s×2 に向上している．6 次元トーラス／メッシュ構造のトポロジや基本的な機能に変更点はない. しかし, Tofu1 に比べ, Tofu2 には不可分操作，セッションモード制御キュー，およびキャッシュインジェクション機能が追加されている点が異なる．不可分操作は InfiniBand にも搭載されているが，InfiniBand 規格で保証されている不可分性は同一 Channel Adapter（CA）上での操作のみであり，他の CA，プロセッサ，入出力装置のメモリアクセスを含めた不可分性のサポートはオプショナルとなっている[10]．このため，ローカルメモリアクセスであっても，不可分性を保証するためには同一 CA を経由したアクセスが必須となり，ローカルとリモート不可分操作のレイテンシはほぼ同等となる[11]．一方，Tofu2 はプロセッサと同一チップ内に統合されているため，Tofu2 とプロセッサの不可分操作は相互に排他となることが保証されている. よって，ローカルメモリに対するアトミック操作はプロセッサで行うことができ，レイテンシの削減に有利である．また，InfiniBand では不可分操作の対象となるのは 64bit 長のデータのみであるが，Tofu2 では 32bit 長のデータの操作もサポートしている．

4. 実装

まず，従来の Tofu1 版の ACP 基本層の実装の概略図を図 1 に示す. Tofu 版 ACP ライブラリはアプリケーション側のメインスレッドとは別に通信スレッドを持つ．アプリケーションから呼ばれた ACP 関数のうち, 通信に関係するものはメインスレッドと通信スレッドで共有されたコマンドキューにエントリを書き込む．通信スレッドはコマンドキューを読み取り, 順次対応する命令を Tofu ライブラリに対して発行する．通信スレッドは，自プロセスの発行する命令だけでなく，他プロセスから委譲された命令も処理する．通信スレッドへの委譲の対象となるのは，ハードウェアに実装されていない機能である不可分操作，およびリモート間コピーである．

次に Tofu2 版の概略図を図 2 に示す．Tofu2 では不可分操作機能がハードウェアに実装されているため，不可分操作は ICC 内で完結させることができる．このため，リモート側の通信スレッドへ操作を委譲させる必要がなくなり，レイテンシの削減が期待できる．本論文では，今まで不可分操作をソフトウェアで実現していた部分を Tofu2 への不可分操作の要求をするように変更した．ただし，Tofu2 ではリモート間コピー機能が存在しないので，この命令に限ってソフトウェア実装を使用しなければならない. よって，通信スレッド内の委譲の待ち受け機構自体は残されている．

5. 性能評価

5.1 評価環境

Tofu2 版の評価には富士通株式会社内の PRIMEHPC FX100 試験機を使用した．Tofu2 のキャッシュインジェクション機能は有効とした．試験機の諸元は表 1 に示した通りである．

表 1 評価環境諸元

マシン名	PRIMEHPC FX100
CPU	SPARC64 XIfx
メモリ	32GB
ネットワーク	Tofu インターコネクト 2, 12.5GB/s
OS	Linux version 2.6.32
コンパイラ	富士通 C/C++ コンパイラ Version 2.0.0
MPI	富士通 MPI ライブラリ Version 2.0.0

5.2 評価対象と方法

実装ごとの性能比較のため, Tofu2 上で評価する通信ライブラリは 3 種類用意した．1 つ目は従来の Tofu1 版とほと

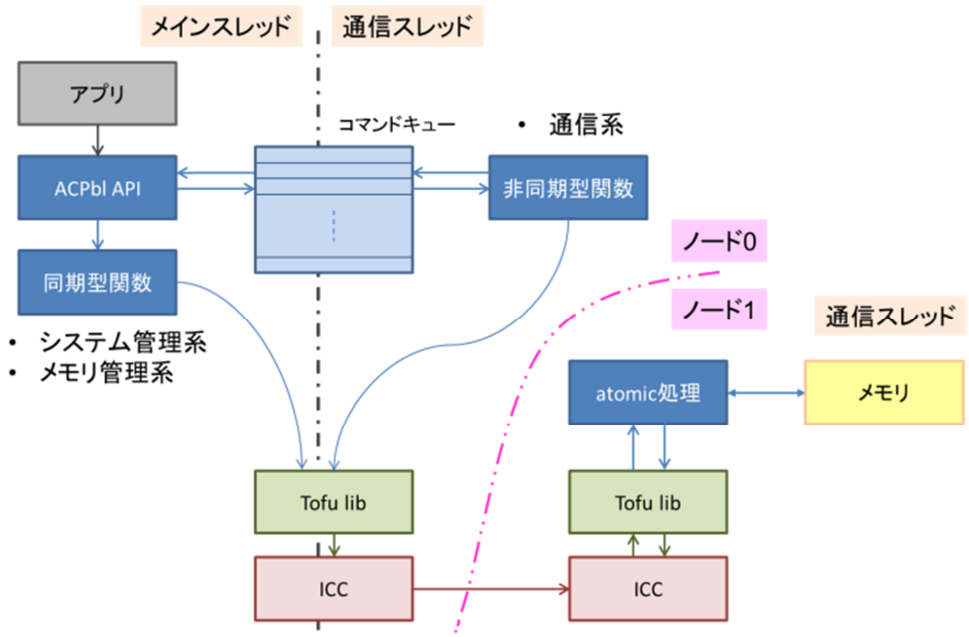

図 1 Tofu1 版の概略図

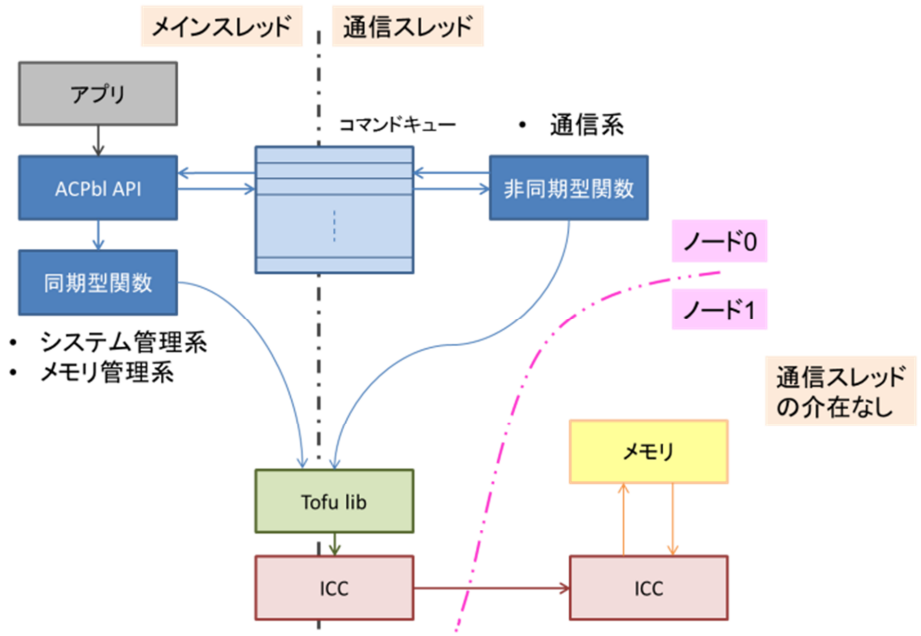

図 2 Tofu2 版の概略図

んど同じ ACP 基本層であり，不可分操作がソフトウェア実装されている．Tofu1 版との差分は，下位通信ライブラリである Tofu ライブラリとのインターフェースの変更点に追従したのみである．2 つ目は不可分操作をハードウェア化したものである．ハードウェア化はローカルデータ・リモートデータの両方に対して行った．ローカルデータに対する操作は Tofu2 を経由しないソフトウェア実装のほうがハードウェア実装より高速であると予想される．ソフトウェア実装とハードウェア実装の性能と比較するため，あえてローカルデータに対するアクセスもハードウェア化してある．3 つ目は ACP 基本層ではなく，富士通 MPI である．富士通 MPI の Tofu2 版は，拡張 RDMA インターフェースに不可分操作のための API が追加されている．富士通 MPI は ACP と異なりシングルスレッドで動作しているため，通信スレッドの影響を排除できる．このため，純粋なハードウェアレイテンシと，現時点でのスーパーコンピュータ製品で使われる通信ライブラリで発生するソフトウェアレイテンシの合計値の指標とすることができる．

また，上記ライブラリの通信性能の評価とは別に，プロセッサで完結する不可分操作単体，および通信スレッドの

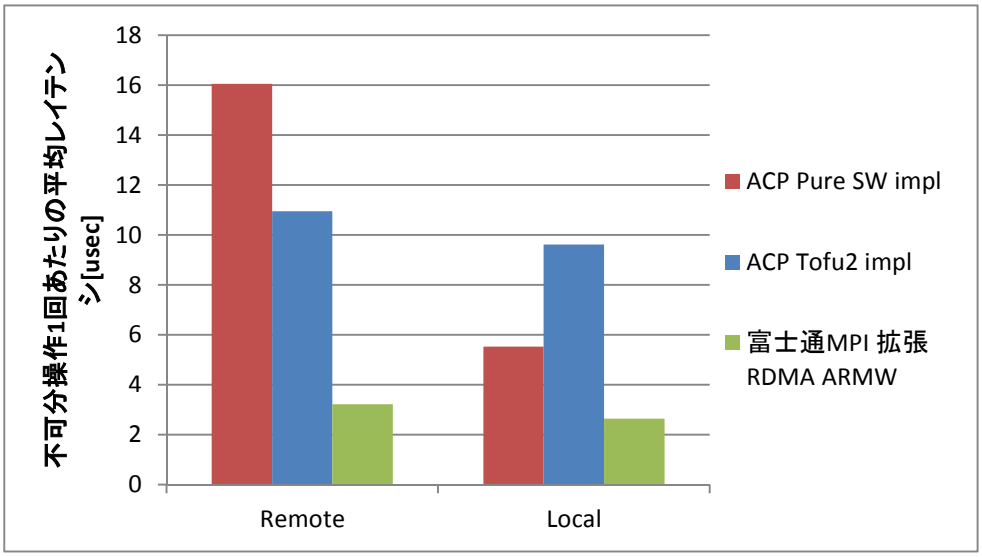

図 3 不可分操作とその完了確認 1 回あたりの時間

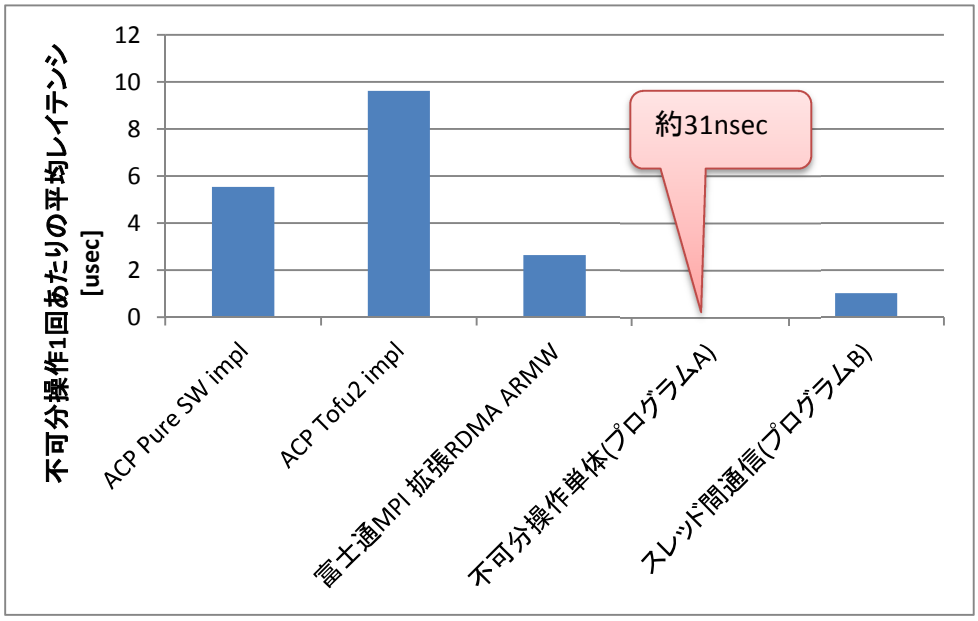

図 4 ローカル変数に対する操作のコスト

影響という個別の要素のコストを評価するため，上記ライブラリの 1 つ目の ACP ライブラリから，不可分操作のソフトウェア実装に使用した不可分操作関数単体を抜き出したプログラム A，およびこのプログラムを，不可分操作を依頼するスレッドと実際に不可分操作を行うスレッドの 2 つに分けたプログラム B を用意した．依頼の機構は，Compare and Swap 命令によるロックを取り合い，状態を表す 4 バイトの共有変数を排他的に監視あるいは書き換えることで実現したものである．共有される変数は ACP ライブラリにおけるコマンドキューより大幅に簡易であるが，基本的な枠組みは同じである．通信ライブラリの評価プログラムは，ランク 0 がリモートあるいはローカルの 4 バイトデータに対して不可分加算を行い，完了を待ち合わせる操作を 1000 回繰り返すものとした．ACP 版は acp_add4 関数と acp_complete 関数，富士通 MPI は FJMPI_Rdma_armw 関数と FJMPI_Rdma_poll_cq_ret_data 関数を使用し，極力同じ動作となるように構成した．また，個別要素の評価プログラムは，通信ライブラリの不可分操作の評価と同様に，4 バイト変数に対する不可分な加算を 1000 回行うものとした．

5.3 評価結果

3 種類の通信ライブラリの，Tofu2 上での不可分操作の性能を計測した結果を図 3 に示す．ここで示されているのは，不可分操作と完了確認の組 1 回にかかる時間の平均値である．凡例にある ARMW は Atomic Read Modify Write の略である．また，図 4 には 3 種の通信ライブラリのローカル変数に対する不可分操作 1 回あたりのコストと，性能に影響

する個別要素のコストを比較した結果を示す.

6. 考察

まず，Tofu2 上の ACP の 2 つの実装に関して分析する．リモート側操作に関してソフトウェア実装とハードウェア実装を比較すると，事前の期待通り約 5.1 マイクロ秒の改善が見られる．不可分操作の委譲機構を撤廃したことで，リモート側のスレッドにおける処理が不要になり，性能が向上している．また，ローカル側操作に関してはソフトウェア実装がハードウェア実装に比べ約 4.1 マイクロ秒高速である．ハードウェア実装のリモート側操作とローカル側操作を比べると，Tofu 通信のレイテンシと思われる差分を除けば同等レベルであるから，ローカル側では Tofu2 を経由しないことに起因してソフトウェア実装の性能が実現していることがわかる．SPARC64XIfx と Tofu2 では，プロセッサとインターコネクトの不可分操作は相互に不可分性が保証されているため，ACP 基本層のハードウェア実装とソフトウェア実装は排他の関係にない．このため，リモート側操作はハードウェア実装，ローカル側操作はソフトウェア実装というように共存させることができ，既存機能の性能を落とすことなくリモート側性能を向上することができる．

次に，ACP の 2 つの実装と富士通 MPI の拡張 RDMA インターフェースを比較すると，2〜5 倍の性能差が見られる．ACP では通信スレッドが存在するため，MPI に比べレイテンシが不利である．ローカル側操作の ACP ソフトウェア実装と富士通 MPI を比較すると約 2.9 マイクロ秒の差があるが，最低限のスレッド間通信を再現したプログラム B は 1 回あたり約 1 マイクロ秒のレイテンシである．プログラム A でプロセッサの行う不可分操作そのもののコストは約 31 ナノ秒であることがわかっているので，プログラム B のコストはほとんどがそのプログラムの仕組み自体により起きていると見なせる．実際の ACP はプログラム B より複雑なコマンドキューを共有しており，レイテンシは単純に通信スレッドとのスレッド間通信のみに起因するのではなく，キューの共有によってコストが上乗せされている．

7. 今後の課題

今回のハードウェア化により確かにリモート操作レイテンシが改善されたが，既存の通信ライブラリに比べ数倍の差がでていることから，より改善の必要がある．今後の性能向上のためには，通信スレッド依存の削減のみならず，ライブラリ内部で保持する情報の設計にも再考が必要である．富士通 MPI はソフトウェアの階層構造が ACP 基本層に比べ厚いため，ACP 基本層から通信スレッドを取り除いた時のソフトウェアレイテンシは，富士通 MPI の拡張 RDMA インターフェースを下回る可能性が高い. ACP では，ライブラリ自体の省メモリ性も重要である．不可分操作をハードウェア化したため，委譲処理用に通信スレッドが保持しているバッファの容量を今後削減できる見込みである．我々のハードウェアとライブラリ実装によって，ローカルアクセスの排他制御を高速に行う見込みが立ったため，単なるデータ参照にとどまらず，排他制御の局所性を活かしたプログラムの記述が可能となった．ACP ライブラリの最適化が進み次第，アプリケーションやグローバルデータ構造の性能評価に移る予定である．

8. まとめ

エクサスケールで想定される多数のノードを持つシステムにおいて，分散データ構造の排他制御の低遅延化は重要であり，その鍵はハードウェア実装された不可分操作である．ACE プロジェクトで過去我々が検討した，インターコネクトが持つべき不可分操作機能を実装した新しいハードウェアである Tofu2 に，ACP ライブラリの基本層を移植した．Tofu2 の機能を利用し，不可分操作をハードウェア化した．性能測定の結果，リモート操作はハードウェア化により従来のソフトウェア実装に比べてレイテンシが改善したことが確かめられた．ローカル操作もハードウェア化した結果，ローカル側操作とリモート側操作の性能はほぼ同等となった．ローカル操作は，プロセッサのみで完結するソフトウェア実装のほうが高速であった．Tofu2 はプロセッサとインターコネクトが緊密に連携しているため，リモート操作は Tofu2，ローカル操作はプロセッサというように混在が可能であり，それぞれ性能の良い方を採用できる点が InfiniBand に比べ優れている．一方，富士通 MPI の拡張 RDMA インターフェースに比べると数倍の差があり, 本来の性能を未だに引き出せていないと考えられる．このレイテンシはスレッド間通信およびスレッド間共有データ構造に起因するので，改善する必要がある．今後はレイテンシ改善に向けて，最低限度の機能以外通信スレッドに頼らない実装および制御情報の最適化を検討する．

参考文献

1) ACE Project, http://ace-project.kyushu-u.ac.jp/main/jp/index.html
2) 住元真司, 安島雄一郎, 佐賀一繁, 野瀬貴史, 三浦健一, 南里豪志: エクサスケール通信向け ACP スタックの設計思想, 情報処理学会研究会報告, Vol. 2014-HPC-143, No. 8, pp. 1-7, 2014.
3) 片側通信による,グローバルデータ構造の効率的な操作方法の検討, Vol. 2012-HPC-133, No. 7, pp. 1-8, 2012
4) 安島雄一郎, 秋元秀行, 安達 知也, 岡本 高幸, 佐賀 一繁, 住元 真司, 三浦 健一: グローバルデータ構造のためのメモリ管理モデルの検討, Vol. 2013-HPC-140, No. 42, pp.1-6, 2013
5) 安島雄一郎, 秋元秀行, 岡本高幸, 三浦健一, 住元真司: 非同期グローバルヒープの提案と初期検討, Vol. 2013-HPC-138, No. 10, pp. 1-6, 2013.
6) 安島 雄一郎, 佐賀 一繁, 野瀬 貴史, 三浦 健一, 住元 真司: ACP 基本層の設計思想とインターフェース, 情報処理学会研究会報告 Vol. 2014-HPC-143, No. 9, pp. 1-6, 2014.
7) 佐賀 一繁, 安島 雄一郎, 野瀬 貴史, 三浦 健一, 住元 真司: ACP 基本層の実装と初期評価, 情報処理学会研究会報告, Vol.

2014-HPC-143, No. 10, pp.1-6, 2014.
8) Ajima, Yuichiro, et al. "Tofu Interconnect 2: System-on-Chip Integration of High-Performance Interconnect." Supercomputing. Springer International Publishing, 2014.
9) Ajima, Yuichiro, et al. "The Tofu Interconnect." High Performance Interconnects (HOTI), 2011 IEEE 19th Annual Symposium on. IEEE, 2011.
10) InfiniBand Trade Association. "InfiniBand? Architecture Specification Volume1 Release 1.2.1." InfiniBand Trade Association, 2007.
11) 秋元秀行, 三浦健一, 岡本高幸, 安島雄一郎, 住元真司: InfiniBand Atomic Operation の性能評価, 情報処理学会研究会報告 Vol. 2012-HPC-133, No. 8, pp 1-6, 2012.