Vol.2015-HPC-148 No.33
2015/3/3

InfiniBand による ACP 基本層の実装と評価

森江 善之[1,5,a)]　南里 豪志[1,5,b)]　安島 雄一郎[2,5]　本田 宏明[1,5]　曽我 武史[4,5]　小林 泰三[3,5]
住元 真司[2,5]

概要：ACE (Advanced Communication for Exa) プロジェクトでは，省メモリかつ低遅延な低レベル通信ライブラリ ACP (Advanced Communication Primitives) の開発を実施している．今回は，HPC 分野で幅広く利用される InfiniBand を用いて，ACP 基本層を実装した．InfiniBand での ACP 基本層の実装方法の報告を行う．また，実装した ACP 基本層のメモリ使用量と通信性能の評価を行った．今回の評価では，InfiniBand の接続資源がメモリ使用量の多く占めることがわかった．また，初期実装の段階で中メッセージサイズ以上で Open MPI と同等の通信性能を示し，最大 20%の性能向上を示した．また，小メッセージサイズでの通信性能の問題を確認することが出来た．

キーワード：エクサスケール，低レベル通信ライブラリ，InfiniBand

Implementation and Evaluation of ACP Basic layer

Yoshiyuki Morie[1,5,a)]　takeshi nanri[1,5,b)]　yuichiro ajima[2,5]　honda hiroaki[1,5]　takeshi soga[4,5]
taizo kobayashi[3,5]　shinji sumimoto[2,5]

Abstract: ACE (Advanced Communication for Exa) project is developing ACP (Advanced Communication Primitives) that is a low level communication library. In this paper, ACP basic layer is implemented on InfiniBand that is important on HPC area. This paper reports the method of implementation, its memory consumption and its communication performance. The connection resource is most memory consumption in this implementation. The performance of ACPbl is nearly equal to Open MPI. Maximum performance improvement is about 20 % in midium of message size. Problem of performance in small message size also was reported.

Keywords: Exa scale, Low level communication library, InfiniBand

1. はじめに

エクサスケール時代における大規模並列計算システムでは，さらにプロセッサのメニーコア化が進むと考えられる．一方，各プロセッサコアあたりのメモリ量の増加は電力対性能比の制約が厳しいことにより望めない．このような制約の中，大規模並列計算システムでは，通信ライブラリの設計を行う必要がある．

このため，大規模並列計算システムでは，省メモリかつ低遅延な通信ライブラリを実現することが課題となる．そこで，我々は，ACE (Advanced Communication for Exa)

[1] 九州大学情報基盤研究開発センター
Kyushu university, Reserch Institute for information techonology, 6-10-1, higashi-ku, fukuoka, Japan
[2] 富士通株式会社 次世代テクニカルコピューティング開発本部
Fujitu Limited., Next Generation Technical Computing Unit
[3] 帝京大学
Teikyo Uniersity
[4] 九州先端科学技術研究所
Institute of Systems, Information Technologies and Nanotechnologies (ISIT)
[5] 独立法人科学技術振興機構　戦略的創造研究推進機構
Japan Science and Technology Agency (JST), Core Research for Evolutionary Science and Technology (CREST)
[a)] morie.yoshiyuki.404@m.kyushu-u.ac.jp
[b)] nanri@cc.kyushu-u.ac.jp

プロジェクト [4] においてこの課題に対応する低レベル通信ライブラリ ACP (Advanced Communication Primitives) の開発を行う [1]．この ACP ライブラリは，低レベルな通信を抽象化する基本層と，基本層の上に実装される中間層で構成されており，中間層はコミュニケーションライブラリおよびデータライブラリというサブライブラリで構成される．現在 ACP 基本層の仕様策定を終え [2]，その実装を公開した．

この ACP 基本層は，Tofu インターコネクトと UDP スタックを用い実装され，初期評価が実施された [3]．一方で，HPC 分野で幅広く利用される InfiniBand による実装および評価はまだなされていない．そこで，本稿では，InfiniBand 向けの ACP 基本層の実装を行い，その評価を実施する．

本稿の構成は以下のようになっている．まず，2 節で ACP 基本層の説明を行う．次に，3 節で InfiniBand による ACP 基本層の実装を述べる．4 節でメモリ使用量について評価を行い，5 節で ACP 基本層の性能評価実験について述べる．最後にまとめと今後の課題を述べる．

2. ACP 基本層

ACP 基本層では，グローバルメモリアドレスを用いた片側通信によるグローバルメモリアクセスを可能とする．従来の分散メモリ型の計算ノード間の片側通信の発行は，送信元と宛先のいずれかのアドレスがローカルメモリに存在するランクがデータの転送を制御する必要があった．しかし，ACP 基本層のインターフェースでは，送信元および宛先ともに引数として ACP が提供するグローバルメモリアドレスを使用することができ，送信元でも宛先でもない第三者のランクがデータ転送を制御できる．これにより，一時的なデータの中継地になる場合など無駄なデータの移動や同期の削減を可能とする．

グローバルメモリアドレスは対象となるメモリ領域をそのメモリ領域を持つランクが ACP ライブラリに登録することで，発行可能となる．リモートランクへアクセスする際は，リモートランクにそのローカルメモリを登録させる必要がある．

提供するグローバルメモリアクセスでは，さらに実行順序を指定することができる．グロバールメモリアクセスの呼び出し時に実行順を各グローバルメモリアクセスハンドルで指定する．呼び出しグローバルメモリアクセスは指定グローバルメモリアクセスハンドルおよびそれ以前に発行されたされたグローバルメモリアクセスがすべて完了した後に開始される．

3. InfiniBand による ACP 基本層の実装

3.1 基本設計

InfiniBand による ACP 基本層の実装では，RDMA (Remote Direct Memory Access) を用いて通信を行う．このため，本実装では，各プロセスとの接続に RC (Reliable Connection) を用い RDMA を実行可能とした．各ランク間の接続に関しては，ACP ライブラリの初期化時に全ランク間で QP (Queue Pair) を用意し，初期化が終了する際には完了させる．この接続は，ACP ライブラリの終了処理時が開始するまで解放しない．

ACP 基本層の構造は，Tofu インターコネクトおよび UDP スタックと同様にメインスレッドと通信スレッドの 2 スレッドで構成される．メインスレッドでは ACP 基本層の関数の呼び出しに対する処理を行う．通信スレッドは ACP ライブラリの通信関連の処理全般を実行する．両スレッドの間にはコマンドキューとキューポインタが配置され，メインスレッドはコマンドキュー経由で通信スレッドに処理を依頼する．コマンドの進捗はキューポインタにより管理する．

メインスレッドで ACP の関数はそれぞれ処理がなされる．まず，同期関数で処理が直ちに完了する関数，または単に状態の変化を待つのみの関数は，通信スレッドに処理を依頼すること無くメインスレッドのみで処理を行う．次に，非同期関数では，コマンドキューにコマンドをエンキューし，このキューエントリを特定できるハンドルを返り値として呼び出しもとに制御を戻す．通信スレッドでは，メインスレッドにより生成されたコマンドに従い，処理を実行する．

また，ACP 基本層では，自分が直接アクセスできるメモリ領域では無い第三者がグローバルメモリアクセスを発行することが可能である．しかし，InfiniBand では，第三者ランクがリモートランク間のデータ転送を実施する機能はない．そこで，これを ACP 基本層にてソフトウェア的に実装する必要がある．このため，通信スレッド上にリモートコマンド受信バッファを置く．ここに，他ランクからのコマンドを受け取る．そのコマンドを受け取ったランクはそのコマンドの内容に従い，適宜処理を行う．

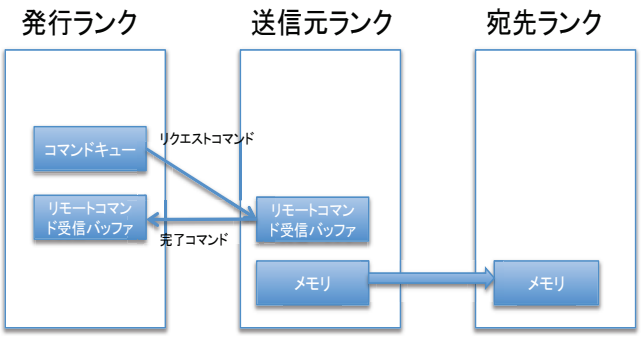

図 1 リモート間コピーの実装
Fig. 1 Implementation of remote to remote copy

第三者のグローバルメモリアクセスを実施する場合は，

図 1 のようになる．まず，第三者ランクがコマンドキューからコマンドを受け取り，そのコマンドを送信元ランクへ渡す．送信元ランクはリモートコマンド受信バッファでそのコマンドを受け取り，解釈し，宛先ランクへのリモートメモリアクセスを発行する．送信ランク側でリモートメモリアクセスの完了を確認したら，発行ランクのリモートコマンド受信バッファへ完了コマンドを送る．発行ランクは，完了コマンドを見て第三者グローバルメモリアクセスを完了させる．

このリモートコマンド受信バッファは各ランクごとに用意するとメモリ消費がランク数に比例するため，エントリの上限を決め，全ランクで共有して使用する．今回の実装では，4096 エントリとする．

通信スレッドは，常にコマンドキューの先頭やリモートコマンド受信バッファ，通信デバイスの状態を QP を通して監視している．コマンドキューやリモートコマンド受信バッファに新たなコマンドがエンキューされたときや通信デバイスの状態に変化があれば，そのコマンドや通信デバイスの状態に沿った動作を実施する．図 2 に ACP 基本層の構造を示す．

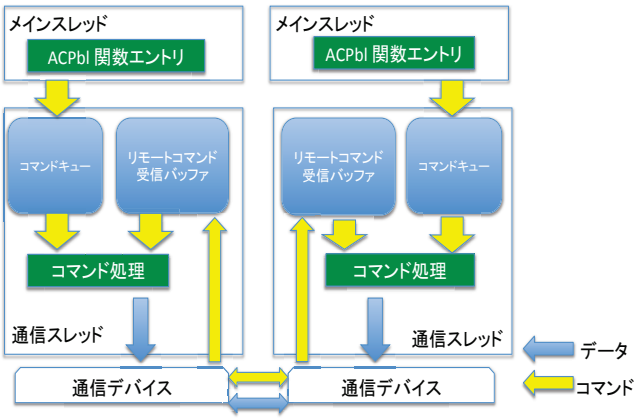

図 2　ACP 基本層の基本構造
Fig. 2　Basci Archtecture

3.2　コマンド

通信スレッドにおいて処理を行うコマンドのフォーマットを表 1 に示す．コマンドキューとリモートコマンド受信バッファは同一のコマンドを使用しており，そのエントリ数はともに 4096 とした．1 エントリあたりデータサイズは 120 B となるため，コマンドキューとリモートコマンド受信バッファのデータサイズは，480 KB となる．

3.3　グローバルメモリアドレスとアドレス変換キー

ACP 基本層においてグローバルメモリアドレスは，アドレス変換キーと論理アドレスをもとに生成される．このアドレス変換キーは，登録するメモリの先頭論理アドレス，

表 1　コマンドフォーマット
Table 1　Command format

コマンド	構成 (Bytes)
基本	ランク (4)，コマンドタイプ (4) ハンドル (8)， オーダハンドル (8)，WR_ID(8))，ステータス (8)， 宛先グローバルメモリアドレス (8) 送信元グローバルメモリアドレス (8) リモートコマンド受信バッファの先頭 (8) リモートコマンド受信バッファの最後尾 (8) コマンドの状態 (8) リプライデータ (8) リモートコマンド受信バッファの有効フラグ (16)
COPY	基本 (104)，サイズ (8)
CAS4	基本 (104)，データ 1 (4)，データ 2 (4)
CAS8	基本 (104)，データ 1 (8)，データ 2 (8)
Atomi4	基本 (104)，データ (4)
Atomi8	基本 (104))，データ (8)

メモリサイズ，このメモリにアクセスする通信デバイスを示すカラー番号をもとに生成される．アドレス変換では，論理アドレスをグローバルメモリアドレスに変換するが，グローバルメモリアドレスの資源は 64 ビットと有限である．このため，他に情報を管理するための識別子を用意し，参照出来るようにする．その情報を登録するテーブルを用意する．これより，登録したメモリ領域の先頭アドレスやサイズをまとめて識別子で管理でき，グローバルメモリアドレスの資源を有効に使用出来る．

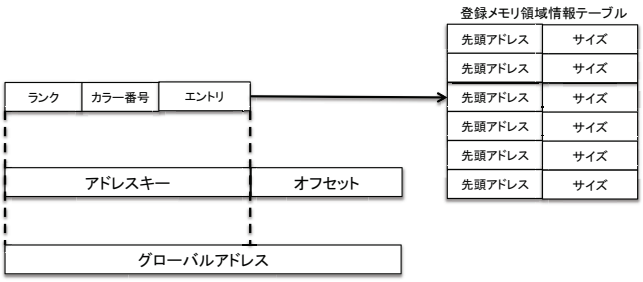

図 3　グローバルメモリアドレスとアドレスキー
Fig. 3　Global memory address and address key

以上から，図 3 のようにメモリ領域を所持するランクの情報，カラー番号，登録メモリ領域情報識別子を連結してアドレス変換キーを作成する．登録メモリ領域識別子は登録メモリ領域情報テーブルを指す．グローバルメモリアドレスは，アドレス変換キーと登録メモリ領域の先頭アドレスからのオフセットを連結したものとする．表 2 に InfiniBand のグローバルアドレスの構成を示す．

また，メモリ領域を登録し，グローバルメモリを生成した場合は，ACP 基本層はこれを他ランクに通知しない．これは，すべてのランクが対応するメモリ領域へアクセスするとは限らないためである．そのような場合はブロードキャストを実行することになり，通信ライブラリの性能を

表 2　グローバルアドレスの構成
Table 2　Global address.

用途	ビット幅
ランク番号	21
カラー番号	1
アドレス変換機識別子	8
オフセット	34

悪化させる．さらに，それらの情報はそのメモリ領域へのアクセスを行わないランクでは，メモリを不必要に消費させることになる．このため，グローバルメモリアドレスはACP 中間層および ACP 基本層のユーザにより必要とされるランク間で交換される．

この時，最初のグローバルメモリアドレスを交換するためにグローバルメモリアクセスが可能なメモリ領域が必要になる．このメモリ領域のことをスターターメモリと呼ぶ．スターターメモリ領域は，ACP の初期化時に生成される．この時，InifiniBand でリモートメモリにアクセス可能とするための情報の交換を終え，その初期化終了時には，全ランクが互いにスターターメモリにアクセス可能となるようにする．スターターメモリのサイズは設定が可能で，基本層のユーザや中間層設計者が必要なメモリ量を設定することで省メモリ化を実現する．

3.4　登録メモリ領域情報テーブルのキャッシュ

InfiniBand 向けの ACP 基本層では，リモートメモリアクセスが発生した場合に RDMA 機能を用いる．この時，InfiniBand では対象となるリモートメモリのアドレス 64 ビットだけでなく，さらに そのメモリ領域の rkey 32 ビットが必要となる．よって，グローバルメモリの資源は有限であるため，rkey も登録メモリ領域情報テーブルに加える．しかし，テーブルは登録したランクしかアクセスできない．リモートランクはグローバルアドレスしか持たないため，識別子がアクセスする先が無い．今回の実装では，この問題に対応するため，登録メモリ領域情報テーブルをキャッシュする方法を導入する．しかし，全ランクでキャッシュを実施すると，メモリ消費量が大きくなるだけなく通信性能にも影響が出る．このため，通信の発生するランクだけにこのテーブルをキャッシュすることする．

具体的には，対象メモリ領域にリモートランクがアクセスする際に，メモリ領域を登録した側のランクにある登録メモリ領域情報テーブルをその領域にアクセスするランク側へコピーする．登録メモリ領域情報テーブルは，初期化時に全ランクでアクセス可能となるよう処理を行う．また，キャッシュ出来るテーブルの最大数は 1024 個とする．それ以上のランクとアクセスする場合は，古いテーブルを削除し，新しいテーブルをコピーしてくる．キャッシュできるテーブルの最大数を決めているため，ランク数に比例して使用メモリ量が増加することが無い．

このリモートのキャッシュの内容は，メモリアクセスが発生する時点において登録メモリ領域の内容と一致していないといけない．ローカルランクにあるメモリ領域の登録を解除し，新たに登録を行った場合，同じ識別子が割り付けられる場合がある．この時，対象となるグローバルメモリへのアクセスが発生し，キャッシュが更新されていなければ，同じ識別子であるにも関わらず，古いメモリ領域にアクセスすることになる．

この問題を回避するため，メモリ領域が解除され，新たに異なるメモリ領域が同じ識別子で割り付けられた場合に自分のテーブルのキャッシュを持つリモートランクへキャッシュの破棄を促すリクエストを発行する．リモートランク側では破棄を要求されたランクのテーブルのみを破棄する．更新ではなく破棄とするのは，以後新たに登録されたメモリへアクセスするとは限らないからである．破棄された後，再びメモリアクセスが発生した際に新たにテーブルをキャッシュする．

4.　メモリ消費量

まず，InfiniBand 上に ACP 基本層を実装した際のメモリ消費量について評価を行う．本評価では，エクサスケール時代の計算機を想定するため，100 万ランクでの実行を仮定する．表 3 に 1 ランクごとに必要なメモリ消費量を示す．

今回の実装で最もメモリ消費量が大きいのは InfiniBand の接続資源である QP でその値は 160 MB となる．これは，InfiniBand では，RC でランク間を接続する場合，QP が接続するランクごとに必要となるからである．また，QP 1 個のデータサイズが 160 B と比較的大きいことも原因である．また，スターターメモリにアクセスするためのリモートアドレスと rkey のテーブルも全ランクの情報を保持するため，メモリ消費量がランク数に比例し，12 MB となる．これ以外にも登録メモリ領域情報テーブルのキャッシュの状態を記録しておく配列もランク数に比例するため，メモリ消費量がそれぞれ 1MB と大きくなる．しかし，QP に対して情報が少ないので，問題とはなっていない．さらに，情報量が 1 ビットで済むので，メモリ消費量を削減できる．

また，登録メモリ領域の情報テーブルのキャッシュの容量も大きい．全ランク分の容量を用意すると 10 GB 必要となってしまう．このため，今回の実装では，保持出来るテーブルの最大数 1024 とし，キャッシュには 10 MB 以上使用しないようにした．これは，すべてのランクと直接通信することはほとんどないと考えられるためである．また，コマンドキューとリモートコマンド受信バッファもランク数に依存しない最大エントリ数を与えているため，ランク数が影響する資源と比べてメモリ消費量が少ない．

表 3　メモリ消費量
Table 3　Memory consumption.

用途	容量
コマンドキュー	480KB
リモートコマンド受信バッファ	480KB
登録メモリ領域情報テーブル	約 10KB (40B * 255 個)
登録メモリ領域情報テーブル	10MB (10KB * 1024 テーブル)
のキャッシュ	
相手登録メモリ領域情報 テーブルのキャッシュの 情報元ランク配列	1MB (1B * 1M ランク)
自登録メモリ領域情報 テーブルのキャッシュの 所持先ランク配列	1MB (1B * 1M ランク)
相手登録メモリ領域情報 テーブルのキャッシュ の破棄要求配列	1MB (1B * 1M ランク)
自登録メモリ領域情報 テーブルのキャッシュ の破棄完了通知配列	1MB (1B * 1M ランク)
QP	160MB (10KB * 1M ランク)
CQ	128B
IB のメモリリージョン 登録テーブル	約 12KB(48B * 255)
スターターメモリ用 リモートアドレス + rkey テーブル	12MB (12 * 1M)

5. 性能評価実験

今後の改良を計画するため，現在の ACP 基本層の実装での性能評価を実施した．

5.1 実験概要

本実験では，性能評価の比較対象として Open MPI[6] を用いる．実行プログラムには IMB4.0 (Intel MPI Benchmarks4.0) [5] を用いた．この時，pingpong 処理の時間を計測する．

一方，ACP 基本層では，スターターメモリに対して put，get を処理する時間を計測する．この時，IMB4.0 の実行回数に合わせて ACP 基本層のグローバルメモリアクセスを同数実行する．これらの通信実行の平均時間から通信性能を比較する．

5.2 実験環境

実験環境としては，PRIMERGY RX200 S7 を用いた．計算ノード数は 16 で，各ノードに Intel Xeon プロセッサ E5-2609 (2.40 GHz) が搭載されている．メモリは 8 GB，計算ノード間は InfiniBand QDR スイッチで接続され，そのスループットは片方向 4.0 GB/s となる．

5.3 実験結果

まず，図 4 にそれぞれの実効通信帯域幅を示す．これより，中メッセージサイズにおいて ACP 基本層の方が高速に動作していることが分かった．特に 8 KB のところで性能比が最大となり，約 20 %の性能向上を示した．しかし，128 MB 以降の大メッセージの領域で Open MPI の方が約 10 %実効通信帯域幅が高いことが分かった．

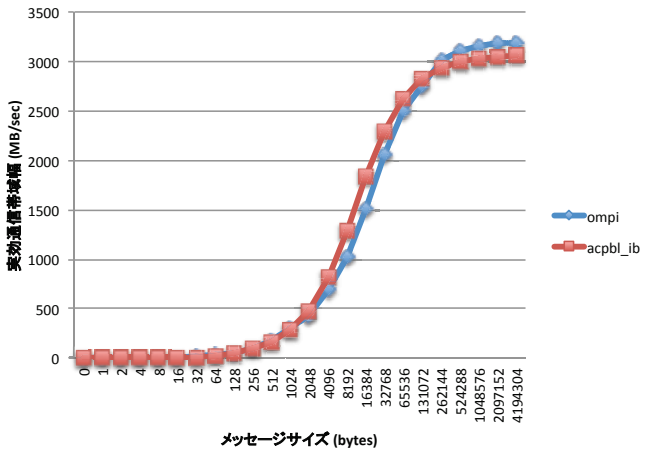

図 4　通信帯域幅
Fig. 4　Communication bandwidth

次に図 5 に通信遅延を示す．メッセージサイズが 1 K までは Open MPI の方が高速であった．特に 64 B 以下のメッセージサイズでは，ACP 基本層の方が 50%前後実行時間が長いことが分かった．これは，通信スレッドにおいてコマンドキューや通信デバイスの状態の監視等を通信の発行と平行して処理を実効することが影響するものと考えている．

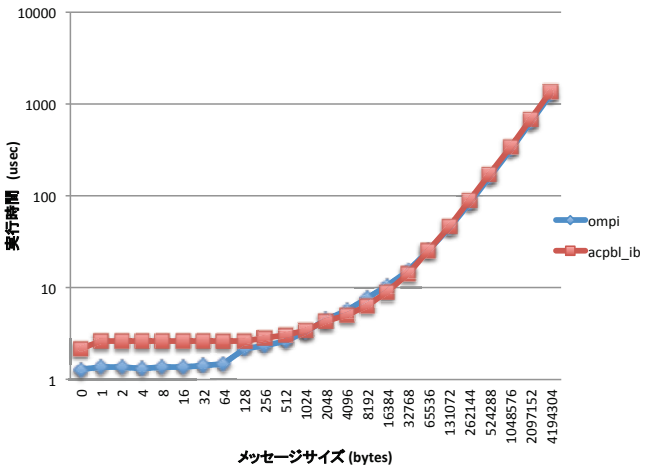

図 5　通信遅延
Fig. 5　Elapsed time

6. おわりに

ACP 基本層の InfiniBand 上への実装概要と通信性能及びメモリ使用量の評価を報告した．

まず，メモリ使用量であるが，項目別メモリ使用量の中で，RC を選択したことに起因する接続資源のメモリ消費量が最も大きいことが分かった．これを解消するため，Dinamically Connection (DC) などを用いて実装することなど検討していく予定である．

次に通信性能について述べる．まず，スターターメモリへのアクセスは，初期実装としては，問題ない通信性能を達成したと考えている．しかし，処理がより複雑なリモート間コピーなどの評価を行っていないため，それらの評価が必要となる．また，スターターメモリへのアクセスに関しても小，大メッセージサイズで低速であることが分かったので，通信スレッドの実装を中心に見直していく予定である．

謝辞　本研究は，科学技術進行機構　戦略的創造研究推進授業 (CREST)「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」研究領域，「省メモリ技術と動的最適化技術によるスケーラブル通信ライブラリの開発」の一部として実施された．ここに記して感謝致します．

参考文献

[1] 住元　真司，安島　雄一郎，佐賀　一繁，三浦　健一，野瀬　貴史，高見　利也，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-143-8 (2014)
[2] 安島　雄一郎，佐賀　一繁，三浦　健一，野瀬　貴史，住元　真司，“ACP 基本層の設計思想とインターフェース”，報処理学会研究会報告，vol.2014-HPC-143-9 (2014)
[3] 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”，報処理学会研究会報告，vol.2014-HPC-143-10 (2014)
[4] ACE Project (online), available from ⟨http://ace-project.kyushu-u.ac.jp/index.html⟩
[5] Intel MPI Benchmarks 4.0 (online), available from ⟨https://software.intel.com/en-us/articles/intel-mpi-benchmarks⟩
[6] Open MPI: Open Source High Performance Computing (online), available from ⟨http://www.open-mpi.org/⟩