Vol.2014-HPC-143 No.10
2014/3/3

ACP 基本層の実装と初期評価

佐賀一繁[†1†2]　安島雄一郎[†1†2]　野瀬貴史[†1†2]　三浦健一[†1†2]　住元真司[†1†2]

ACE（Advanced Communication for Exa）プロジェクトでは，省メモリ，低遅延な通信スタック ACP（Advanced Communication Primitives）の検討を行っている．ACP 基本層は低位通信層でありグローバルメモリ空間と片側通信を特徴とする．今回 ACP 基本層を Tofu インターコネクトと UDP スタックという性質，機能が大きく異なる 2 つの通信機構に実装し初期評価を行った．本報告ではこれらの設計と実装方法，特にグローバルメモリ空間，片側通信，遠隔ランク間データ参照の実現方法，通信機構による実装方法の違い，実装の注意点など説明し，初期評価結果を報告する．

Implementations and Evaluations of ACP Basic Layer

KAZUSHIGE SAGA[†1†2]　YUICHIRO AJIMA[†1†2]　TAKAFUMI NOSE[†1†2]
KENICHI MIURA[†1†2]　SHINJI SUMIMOTO[†1†2]

ACE (Advanced Communication for Exa) project is developing a communication protocol stack named ACP (Advanced Communication Primitives). ACP basic layer is a basic communication layer which uses a concept of global memory and on side communication. We have implemented it on Tofu interconnect and UDP stack, and done an evaluation. In this paper, we focus design and implementation of them, especially for global memory space, one side communication, initiation for data access between remote ranks, and consideration of implementation, and then discuss the result of an evaluation.

1. はじめに

ACE（Advanced Communication for Exa）プロジェクトでは，省メモリ，低遅延な通信スタック ACP（Advanced Communication Primitives）の検討を行っている [1]．ACP は，基本的な通信を担う基本層，ストリーム転送の仕様メモリを最適化するためのチャネルインターフェース，大域的なデータ配置を最適化するグローバルデータ構造コレクションから構成されており，現在 ACP 基本層の仕様策定を終了し提案を行っている [2]．ACP 基本層では従来の通信方式で暗黙的に消費されているメモリを削減するために，グローバルメモリ技術と片側通信技術を採用している．具体的にはグローバルメモリ空間に通信データを配置し，これを各プロセスが片側通信によって低遅延な直接アクセスすることにより，従来の通信方式で暗黙的に使用されていた通信バッファなどのメモリ資源の低減を実現している．

今回 ACP 基本層仕様を検証するために，スーパーコンピュータ「京」や富士通社製スーパーコンピュータ FX10 に採用されている高性能インターコネクト Tofu（Torus Fusion）[3]と UDP スタックという機能仕様の全く異なる 2 つの通信機能上に参照実装を行い初期評価を行った．本論文ではこれらの実装設計，実装方法などを説明し，初期評価結果を報告する．

まず，第 2 章で今回の実装方針について説明し，第 3 章で設計に影響を与える ACP 基本層の特徴を，第 4 章でこの特徴を実現するための基本設計を説明する．第 5 章では実装概略を説明し，デバイスの機能によるに実装の違い，実装上の注意点などを説明する．第 6 章ではメモリ使用量について，第 7 章では初期評価結果を報告し，第 8 章でまとめと将来構想について説明する．

†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 CREST
Japan Science and Technology Agency (JST),
Core Research for Evolutionary Science and Technology (CREST)

2. ACP 基本層の実装方針

新たな通信方式を広く提案するには，仕様の異なる複数の通信デバイスに実装し，動作や効果を確認する必要がある．今回我々が実装対象とした Tofu インターコネクトと UDP スタックは，機能，性格が全く異なる．Tofu は 6 次元メッシュ／トーラス構造の直接網インターコネクトで，双方向 5Gbps と高バンド幅の通信リンクを持ち，RDMA（Remote Direct Memory Access）による低遅延通信機能持つ．また 8 万ノード以上の大規模システムに対応した実績があり，エクサスケール時代の通信方式を検討する素材として適していると考えている．また UDP スタックは，高性能よりも汎用性にすぐれており，様々な通信プロトコルの低位通信層として採用され，また様々な通信ハードウェアでサポートされている．このためその技術も広く知れ渡っており，多くの開発者に実装方法を理解していただく参照実装の素材として最適であると考えている．なお，ACE プロジェクトでは InfiniBand への実装も検討している．

3. ACP 基本層の特徴

本章では ACP 基本層の実装設計で考慮すべき，いくつかの特徴的機能について概要を説明する．

3.1 グローバルアクセスと直接通信による省メモリ化

一般に従来のデータ通信は通信バッファを介して行われ，高性能な HPC システムの通信でも例外ではない．一般的な通信方式では少数の通信バッファを複数の通信先に対して共有するが，HPC システムの通信方式では低遅延化のために通信先毎の通信バッファを持つことが多い．このため HPC システムの通信の方が暗黙的なメモリの使用量が多く，またランク数に比例して消費する実装も多いので，コア数が膨大なると予想されているエクサフロップスシステムに向け問題視されている．ACP 基本層はこの暗黙的に使用されているメモリの低減と低遅延の両立を目標に，グローバルメモリアクセスと片側通信を基にした関数を定義している．グローバルメモリ空間のデータへの参照は，CPU 直接か通信かに係らず，グローバルメモリ参照と呼ばれる．

3.2 グローバルメモリの参照と管理

ACP 基本層において，データをグローバル参照可能にするには，まずそのデータのメモリ領域をグローバルアドレス空間に配置する必要がある．具体的にはデータの先頭論理アドレスとサイズをメモリ登録関数で ACP 基本層に登録しアドレス変換キーを取得する．次にこのアドレス変換キーと参照データのアドレスオフセット（登録した先頭論理アドレスからのオフセット）を引数としてグローバルアドレス取得関数を呼び出すことによりグローバルアドレスが取得できる．このグローバスアドレスを使用すれば，グローバルメモリ参照関数で参照可能となる．

ACP 基本層はこのアドレス変換キーの管理を行う．ACP 基本層では近傍のメモリ領域が連続して登録された場合にマージして 1 つのアドレス変換キーにすることが許されている．ただし，メモリ登録削除の際には登録回数分の登録削除関数の呼び出しが必要である．このため ACP 基本層は同一アドレス変換キーに対する登録回数を管理しなくてはならない．また ACP 基本層にはグローバルアドレスから，そのメモリ登録時に指定したプロパティを取得する関数があるため，グローバルアドレスからアドレス変換キーや登録時プロパティを効率的に参照できる管理構造にしなくてはならない．

3.3 メモリロケーション非依存

ACP 基本層では，実メモリがどのランクに存在していても操作，参照できる柔軟な関数と機能の実現を目指している．このため，グローバルメモリアドレスには実メモリが属するランクを特定できる情報が含められており，参照関数の引数にはランク指定を持たない．これにより遠隔ランク間のデータ転送や遠隔ランク間のアトミック操作の第三者ランクによる起動が容易に記述できる．また実行毎にランク数が変化するアプリケーションでも，実メモリロケーションと参照するランクを関係づけるプログラミングが不要となり，生産性の向上を期待できる．

このように遠隔ランク間のデータ処理を第三者ランクが起動できると，一時コピー用バッファの削減，ネットワークトラフィックの低減，処理時間の短縮，延いては消費電力の低減などに効果を発揮する場合ある．これらの多くは，エクサフロップス時代の HPC システムとして解決すべき課題として挙げられている項目であり，ACP 基本層の利点のひとつとなっている．

4. 基本設計

本章では第 3 章で説明した特徴を考慮した設計について説明する．まず，共通な基本制御構造の設計について説明し，次にメモリロケーション非依存を実現するためのコマンドデリゲーション機能の設計について説明する．なお，Tofu インターコネクト, UDP スタック以外の実装でも適用可能な設計とした．

4.1 基本制御構造

基本的な構造は，メインスレッドと通信スレッドの２つのスレッドで構成している．メインスレッドは関数のエントリとして呼び出し元スレッドの延長として動作し，通信スレッドは通信関連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．基本制御構造を図 1 に示す．

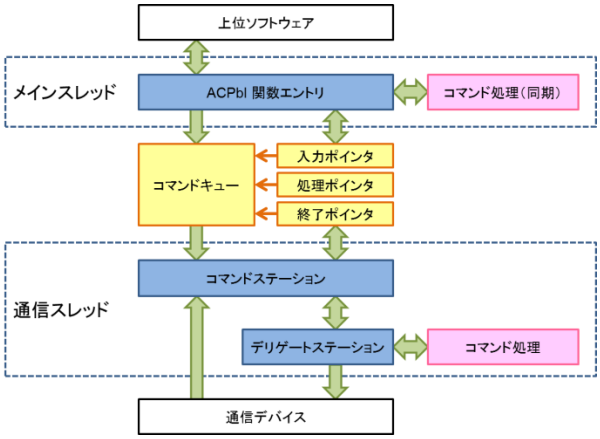

図 1　基本制御構造
Figure 1　Basic Architecture.

同期型関数で処理が直ちに完結する関数，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なくメインスレッドのみで処理を行う．他ランクへのアクセスが必要な同期型関数は，コマンドキューにコマンドをエンキューしてメインスレッドで終了を待つ．非同期型関数では，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元に制御を戻す．なお，エンキュー動作はスレッドセーフのために排他制御する．

通信スレッドは，常にコマンドキューの入力ポインタと通信デバイスの状態を監視しており，コマンドキュー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行う．

4.2 コマンドデリゲーション

第 3.3 節で述べたように，ACP 基本層のグローバルメモリ参照関数は実メモリが存在するランクに非依存な仕様である．しかし，現在のところ第三者ランクが遠隔ランク間のデータ転送をする機能や，遠隔ランクのアトミック操作を行う機能をもつ通信デバイスないと思われる．そこで，これら機能をソフトウェアで実現する設計を行った．具体的には通信スレッド上にデリゲートステーションとコマンドステーションを置く．デリゲートステーションはコマンド処理を実行し，コマンドステーションはコマンドキューからのコマンドの取り出し，キューポインタの管理，デリゲートステーションへのコマンド実行依頼を行う．遠隔ランクでコマンド処理を行う必要がある場合，コマンドステーションはそのランクのデリゲートステーションにコマンドを転送して処理を依頼する．このデリゲートのための通信は ACP 基本層に閉じた通信であり，上位プログラムからは隠蔽される．

自ランクが第三者となる遠隔ランク間操作の場合，どちらのランクに依頼するかはコマンドにより異なる．遠隔ランク間コピー関数の場合はどちらでもよく，通信デバイスの都合のよい方を選択できる．遠隔ランク間アトミック通信の場合ソースランクにデリゲートする．図 2 に第三者ランクによる遠隔ランク間の CAS（Compare And Swap）操作の流れを示す．ランク 0 がディスティネーションであるランク 1（旧データを戻すランク）とソースであるランク 2（データを置き換えるランク）の間のアトミック操作を実行する例である．もし，ディスティネーション（ランク 1）にデリゲートするとランク 1 からランク 2 へ CAS コマンドのデリゲートが必要となり実行開始が遅れる．

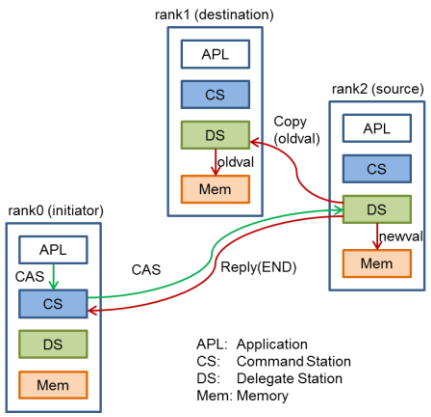

図 2　第三者による遠隔ランク間アトミック操作
Figure 2　Third Party Initiated Remote Atomic Operation.

4.3 アドレス変換キーとグローバルアドレス

ACP の仕様では，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，メモリサイズ，このメモリをアクセスするネットワークインターフェースを示すカラー番号を元にメモリ登録関数で生成される．アドレス変換機構は論理アドレスをグローバルアドレスに変換するが，その変換資源は有限であり管理のための識別子がある．また変換が有効な範囲の指定としてサイズの登録も必要であり，論理アドレスとともに管理される．このため，論理アドレスとサイズはひとつの識別子で管理することができる．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から生成される仕様となっている．以上から，アドレス変換キーをプロセスランク情報，カラー情報，アドレス変換機構のエントリの識別子を連結して表現し，グローバルアドレスはこのアドレス変換キーと指定された論理アドレスと登録されたメモリの先頭論理アドレスの差分（オフセット）の連結で表現するものとした．これを図 3 にしめす．

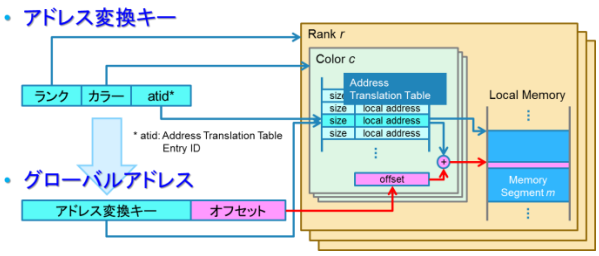

図 3　アドレス変換キーとグローバルアドレス
Figure 3　Address Translation Key and Global Address.

上位層からの関数呼び出しで新たにメモリがグローバル空間に登録されグローバルアドレスが生成されても，ACP 基本層はこれを他ランクに通知しない．各ランクがすべてのグローバルメモリをアクセスすることは少なく，全てのランクが全てのグローバルメモリのアドレスを持つことはメモリの浪費に他ならないためである．また，使用するグローバルアドレスのみ通知しようにも ACP 基本層ではどのグローバルメモリが使用されるか判定はできない．グローバルアドレスは，関数を使用する上位ソフトウェアが必要に応じて他ランクに通知する．ただしコマンドデリゲーションのためのメモリ領域のグローバルアドレスは，ACP 基本層自身が他ランクの ACP 基本層と交換する必要がある．なお 1 ランク／1 ノードの環境では，スタータセグメントと同様に初期化時に 1 番目のユーザ空間セグメントとして登録し，この交換を省略することもできる．

5. 実装

本章では，前章の基本設計を基にした Tofu インターコネクトと UDP スタックへの実装を説明する．

5.1 基本制御構造の実装

Tofu 実装，UDP 実装とも基本的な制御構造は図 1 の設計を採用し，コマンドキューの深さは 4,096 エントリとした．コマンドキューの 1 エントリあたりのサイズは，Tofu: 64bytes，UDP: 80bytes であり，キューのサイズは Tofu: 262Kbytes，UDP: 327KBytes となっている．ハンドルの実装も同じであり，キューのエントリ番号を採用している．

5.2 コマンドデリゲーション機能の実装

コマンドデリゲーションのフォーマットは Tofu 実装と UDP 実装で異なる. Tofu 実装のコマンドフォーマットを表 1 に，UDP 実装のコマンドフォーマットを表 2 に示す.

表 1　Tofu 実装のデリゲーションコマンドフォーマット
Table 1 Delegation Command Formats of Tofu Implementation.

フォーマット（バイト）	構成（バイト）
基本(22)	並列プロセス ID (4) , コマンド(2), ランク(4), ハンドル(2), ステータス(2), リプライデータ(8)
COPY(48)	基本(22), 親ランク(4), 親ハンドル(2), GA_DST(8), GA_SRC(8), サイズ(4)
CAS4(52)	基本(22), 親ランク(4), 親ハンドル(2), GA_DST(8), GA_SRC(8), データ 1(4), データ 2(4)
CAS8(56)	基本(22), 親ランク(4), 親ハンドル(2), GA_DST(8), GA_SRC(8), データ 1(8), データ 2(8)
RMW4(48)	基本(22), 親ランク(4), 親ハンドル(2), GA_DST(8), GA_SRC(8), データ(4)
RMW8(56)	基本(22), 親ランク(4), 親ハンドル(2), GA_DST(8), GA_SRC(8), データ(8)

表 2　UDP 実装のデリゲーションコマンドフォーマット
Table 2 Delegation Command Formats of UDP Implementation.

フォーマット（バイト）	構成（バイト）
COPY(48)	TASK(4), TYPE(4), RANK(4), SEQ(4), PTR(8), DST(8), SRC(8), SIZE(8)
CAS4(48)	TASK(4), TYPE(4), RANK(4), SEQ(4), PTR(8), DST(8), SRC(8), OLD4(4), NEW4(4)
CAS8(56)	TASK(4), TYPE(4), RANK(4), SEQ(4), PTR(8), DST(8), SRC(8), OLD4(8), NEW8(8)
Atomic4(44)	TASK(4), TYPE(4), RANK(4), SEQ(4), PTR(8), DST(8), VAL4(4)
Atomic8(48)	TASK(4), TYPE(4), RANK(4), SEQ(4), PTR(8), DST(8), VAL8(8)
Put(32+)	TASK(4), TYPE(4), RANK(4), SEQ(4), PLEN(8), DST(8), DATA
ACK/NACK/ RETRY(16)	TASK(4), TYPE(4), RANK(4), SEQ(4)
Read(32)	TASK(4), TYPE/ID(4), RANK(4), LEN(4), PTR(8), DST(8)
ReadACK(32+)	TASK(4), TYPE/ID(4), RANK(4), LEN(4), PTR(8), DST(8), DATA
Write(32+)	TASK(4), TYPE/ID(4), RANK(4), LEN(4), PTR(8), DST(8), DATA
END/WriteACK/ EndACK(24)	TASK(4), TYPE(4), RANK(4), reserve(4), SEQ(4)

アトミック操作は，デリゲートステーションから呼ばれる各コマンドの処理の中でインラインアセンブラを使用して実現している. x86_64 システムでは gcc4.2 以降に組込まれている__sync 系関数を使用し, SPARC システムでは同等の関数を作成している．なお，現在の Tofu 実装では第三者ランクによる遠隔ランク間アトミック操作は未実装となっている.

現在の Tofu 実装では, グローバルメモリ参照関数の通信を含め，すべて通信は Tofu の RDMA 機能を基にした通信としている．グローバルメモリ参照関数で依頼された通信は依頼元が獲得したグローバルメモリ間で通信するため，ACP 基本層が暗黙的に取得する通信領域はない．しかし，ACP 基本層間の独自通信では通信領域として暗黙的に獲得される領域がある．ランク間コマンドデリゲーション用の受信バッファである．送信はコマンドキューエントリから直接送信しているため通信専用ではない．現在の実装では各ランクに全ランク分のコマンド受信バッファを獲得している．このコマンド受信バッファはコマンドキューエントリと同じフォーマットで 64byte／ランクである．このため 100 万プロセスにおける総容量は 64Mbytes にもなり, 従来の通信プリミティブのデータ通信用バッファに比べ容量は少ないものの，ランク数に比例して容量が増加するため改善が必要と考えている．なお，将来のインターコネクトで遠隔間操作や遠隔アトミック操作がサポートされれば，この受信バッファは削除または低減することが可能である.

5.3 アドレス変換キーとグローバルアドレスの実装

第 4.3 節の設計通り，アドレス変換キーはプロセスランク情報，カラー情報，アドレス変換機構のエントリ識別子を連結させて実装し，グローバルアドレスはこのアドレス変換キーと登録されたメモリの先頭らの差分（オフセット）を連結させて実装した.

Tofu では，分散環境のジョブ実行をサポートするランタイムシステムがあるため，プロセスランク情報はシステムから得られるランク番号もしくは再起動時に上位ソフトウェアが指定したランク番号とした．またカラー情報はこのグローバルメモリのアクセスに使用する TNI（Tofu Network Interface）の番号を，アドレス変換機構のエントリの識別子は Tofu の STag（Steering Tag）と呼ばれるメモリブロック識別子を使用することにした．だたし，ACP ではグローバルデータ構造コレクションでリモートメモリアロケーションの実現を予定しており [4]，リモートのメモリ管理ポインタを更新する必要があるため，グローバルアドレスの幅を 64bit 以内に収める必要がある．Tofu における各要素の最大値のビット幅の合計は 64bit を上回っており圧縮する必要がある．しかし固定的な圧縮ではその割合がアプリケーションの実行時に求められる各要素の値の割合と合致せず，割合が異なれば実行できるものが実行できなくなる可能性がある．そこでグローバルメモリアドレスのビット

構成は実行時に決定するランク数適応型構成とした．グローバルメモリアドレスは各実行時にのみで有効な値であるため，実行毎にビット幅を変化させても問題はない． 図 4 に Tofu 実装のグローバルメモリアクセス構造を，表 3 にグローバルアドレス構造を示す．

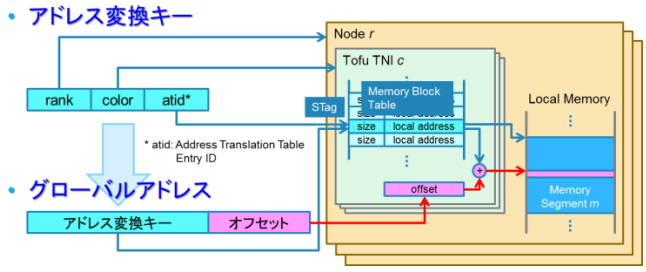

図 4 Tofu 実装のグローバルメモリアクセス

Figure 4 Global Memory Access of Tofu Implementation.

表 3 Tofu 実装におけるグローバルアドレス

Table 3 Global Address in Tofu Implementation.

GMA	Tofu 実装	ビット幅
ランク番号	システムから付与されたランク番号	5 ~ 20
カラー番号	Tofu TNI 番号	2
アドレス変換機構の識別子	Tofu STag 番号	22 ~ 7
オフセット	最大オフセット値（＝ノードのメモリ量）	35

UDP 実装では，Tofu のようなジョブ実行をサポートするランタイムシステムがないため，初期化時のランク番号，プロセス数はアプリケーションの起動時にアプリケーション経由で与えるものとしている．現在の実装ではカラー番号を無視しておりネットワークインターフェースは 1 つしか使用できない．アドレス変換機構のエントリは 16 個に限定しており，内 4 つはスタータメモリとして定義している．現在の ACP 基本層仕様ではスタータメモリは1つとなっているが，将来のグローバルデータ構造コレクションなどのために拡張できることを考慮している．また一般メモリに対するエントリも 1 つに制限しているが，メモリの全領域を登録することができるので試用には問題ない．表 4 に UDP 実装のグローバルアドレス構造を示す．

表 4 UDP 実装におけるグローバルアドレス

Table 4 Global Address in UDP Implementation.

GMA	UDP 実装	ビット幅(bit)
ランク番号	アプリケーションの引数で与えられたランク番号	1 ~
カラー番号	無視	—
ローカルセグメント番号	初期化で定義されるセグメントの番号．うち 4 つスタータセグメント	4
オフセット	最大オフセット値（＝ノードのメモリ量）	1 ~

5.4 通信の送達保障と順序保証

Tofu ハードウェアは通信の送達保障機能，順序保証機能を持ち，特別な処理をしなくとも常に送達保障される．また順序保証については，Tofu ハードウェアの通信コマンドのフラグで必要に応じ順序保証を有効にすることができる．一般に順序保証を行うと通信スループットが低下するため，必要に応じて使い分けることが必要である．

UDP 実装では送達保障，順序保証ともソフトウェアで実現している．送達保障は，受信側がすべてのデータグラムに対し ACK を返し，送信側が ACK のタイムアウトを検出することで通信がロストした可能性を検出する．これを検出したときに再送信を行うことによって送達保障を実現している．なお輻輳によるロストを低減するために，連続してタイムアウトが発生すると再送信間隔を倍増させている（最少 100 μ 秒，最大 100m 秒）．順序保証はすべてのデータグラムに宛先ランク毎のシーケンス番号を付加し，受信側でシーケンス番号が非連続であれば ACK を返さずにデータグラムを破棄することにより実現している．

5.5 その他関数の実装

(1) 初期化処理

初期化処理では，内部管理データ，通信デバイス，通信バッファの初期化，通信スレッドの起動など多くは一般的通信ライブラリと同様の項目であり実装も変わらない．多少異なるのはスタータメモリの設定であり，グローバルセグメント空間を初期化してからでないと初期化できない，初期化直後に通信可能な状態とする必要があるなどの条件の，初期化の最後付近で行っている．なおスタータメモリのグローバルメモリ空間からの登録削除は，終了処理でしかできないので，特別なメモリセグメントとして明確に管理している．

(2) 終了処理

終了処理も一般的な通信ライブラリの終了処理と大きくは変わらない．中途半端な制御状態や通信状態が残っていれば刈取り，初期化または再初期化以降に獲得した全ての資源を解放する．ただし，ACP 基本層は MPI などと異なり終了処理後の再初期化とその後の使用を許しているため，プロセス終了時のシステムによる資源の解放は期待せず，初期化または再初期化以降に獲得した資源は確実に解放している．

(3) 再初期化処理

再初期化処理は ACP 基本層の特徴のひとつである．初期化では，システムが割り当てるランク番号とスタータメモリサイズで初期化するが，再初期化では呼び出し元が指定した任意のランク番号と任意のスタータメモリサイズで初期化を行う．再初期化以前に取得した資源は全て解放する ACP 仕様のため，両実装とも終了処理と初期化処理を連続して実行することで実現している．ただし再初期化では指定されたランクとスタータメモリサイズを使用する．

6. 使用メモリ量

Tofu 実装の主要消費メモリを表 6 に示す．最もメモリ消費が大きいのはランク間コマンドデリゲーションのコマンド受信バッファである．第 5.2 節で述べたように，将来のインターコネクトで遠隔ランク間データ処理，遠隔アトミック操作がサポートされれば，削除されるか容量を低減できる見込みである．また Tofu アドレステーブルも大きい．ランク番号から Tofu アドレスを解決するにはシステムへの問い合わせが必要である．低遅延を実現するために通信毎に問い合わせるのではなく，初期化時に全実行ランクの Tofu アドレスをシステムに問い合わせテーブル化している．キャッシュ型の構造に変更し通信頻度の高い Tofu アドレスのみを持つように変更予定である．

表 5　Tofu 実装の使用メモリ量
Table 5　Memory Usage of Tofu Implementation.

用途	容量
コマンドキュー兼送信／リプライ受信バッファ	262Kbytes (64bytes×4,096entries)
コマンド受信バッファ	64Mbytes@1M ranks (64bytes×ranks)
登録メモリ管理テーブル	40 bytes×登録メモリ数
論理アドレス検索テーブル	16 bytes×登録メモリ数
Tofu アドレステーブル	4Mbytes@1M ranks (4bytes×ranks)

UDP 実装の主要消費メモリを表 7 に示す．コマンドステーションでは，再送信管理のための双方向リストを持ち，使用量は，最大データサイズを 1,408 バイトとした場合，1,504 バイト×64 エントリで固定である．デリゲートステーションでもパケットの再送信管理に固定な 48 バイト×64 エントリの領域を使用し，またコマンドの実行管理に 3,480 バイト×64 エントリの固定領域を使用している．

表 6　UDP 実装の使用メモリ量
Table 6　Memory Usage of UDP Implementation.

用途	容量
コマンドキュー	327Kbytes (80bytes×4,096)
コマンドステーション	96Kbytes (1,504×64)
デリゲートステーション	222Kbytes (1,504×64)
待ち受け IP／ポート	6Mbytes@1M ranks (6bytes×ranks)
シーケンス番号（送信，受信）	8Mbytes@1M ranks (8bytes×ranks)
ランク番号	4Mbytes@1M ranks (4bytes×ranks)

7. 初期評価

すべての関数を使用したテストプログラムを両実装で実行し，同一の結果が得られることを確認した．ただし，Tofu 実装における遠隔ランク間アトミック操作など未実装項目が含まれるテストや，カラー数などシステム構成の違いが影響するテストでは動作結果が異なった．

8. まとめ

ACP 基本層の Tofu インターコネクトと UDP スタック上への参照実装の概要，両者の実装の違い，実装上の注意，初期評価などを報告した．両者は機能，性格がかなり異なる通信デバイスであるが，問題なく実装でき仕様通り動作することが確認できた．また共通の上位プログラムが動作することも確認しており，異なる通信デバイスでの関数互換性も問題ないことを確認できた．しかし Tofu 実装のメモリ使用量は，従来のメッセージパッシング型通信プリミティブよりは少ないものの改善が必要であると考えている．改善を継続する．

将来構想としては，まず ACP 基本層実装の完成度を向上させ，グローバルデータ構造コレクションなど上位層の検討の推進に貢献し，ACP を総合的な省メモリ，低遅延の通信ライブラリとして確立していきたいと考えている．またさら言語との連携をとり，ACP より有用なものにしたいと考えている．

謝辞　本研究は，科学技術振興機構 戦略的創造研究推進事業（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．

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 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，情報処理学会研究会報告 2014-HPC-143-9 (2014).
3)　Y. Ajima, T. Inoue, S. Hiramoto, T. Shimizu, Y. Takagi, “The Tofu Interconnect,” IEEE Micro, vol. 32, no. 1, pp.21-31 (2012).
4)　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，情報処理学会研究会報告 2014-HPC-138-10 (2013).