Vol.2022-ARC-248 No.49
Vol.2022-SLDM-198 No.49
Vol.2022-EMB-59 No.49
2022/3/11

分散共有メモリ型通信機構 MBCF/Linux の基本性能について

大家彩[1]　猪俣雪乃[1]　上村敬志[2]　松本尚[1]

概要：オリジナル並列分散 OS SSS-PC の基幹となる通信同期機構 MBCF の Linux OS 版である MBCF/Linux の開発を行った．高スループットと低レイテンシを両立し，高機能な共有メモリ操作を実現し，ロックフリーの不可分共有メモリ操作が可能な MBCF 通信機構を Linux 上で動かすことに成功した．本稿では，MBCF/Linux の様々な環境におけるスループットとレイテンシといった基本性能測定の結果について述べる．

キーワード：MBCF，MBCF/Linux，分散共有メモリ，通信同期機構，SSS-PC

Basic Performance Reports of Memory-Based Communication Facility in the Linux kernel

AYA OYA[†1]　YUKINO INOMATA[†1]　KEIJI UEMURA[†2]
TAKASHI MATSUMOTO[†1]

Abstract: We have developed MBCF/Linux, which is the Linux OS version of MBCF, which is the mandatory communication synchronization mechanism of our original parallel distributed OS SSS-PC. It achieves both high throughput and low latency and realizes high-performance shared memory operations, and also achieves lock-free atomic shared memory operations on Linux. In this paper, we describe the results of basic performance measurements such as throughput and latency in various environments of MBCF/Linux.

Keywords: MBCF，MBCF/Linux，Distributed Shared Memory，communication and synchronization mechanism，SSS-PC

1. はじめに

オリジナル並列分散 OS である SSS-PC[3] および SSS-CORE[2] の基盤となる通信同期機構として採用していた Memory Based Communication Facility (**MBCF**)[4][5]の Linux OS 版である MBCF/Linux を開発するとともに, MBCF を 64bit アドレスに対応させた MBCF64 を新たに規定して, 64bit アドレスの Linux OS にも対応可能にした．2021 年 12 月の報告[1]において，MBCF の概要を振り返り，64bit アドレス対応のための拡張方式を解説し，Linux 版の開発基本方針と実装時の注意点を述べた．本稿では，紙面と発表時間の都合で説明できなかった MBCF/Linux のスループットとレイテンシの基本性能測定結果について報告する．なお, 64bit アドレス対応の MBCF/Linux であることを明示したい場合は MBCF64/Linux と表記して，従来の 32bit アドレス対応の Linux 版 MBCF であることを明示したい場合には MBCF32/Linux と表記する．なお， SSS-PC 版と Linux 版に共通な特徴や機能に言及する場合には，単に MBCF, MBCF32, MBCF64 と表記する．

2. MBCF の概要

MBCF は Gigabit Ethernet のような高速ネットワークで接続された PC クラスタやワークステーションクラスタにおいてマシン境界を越えた共有メモリアクセス環境を実現するための通信同期機構として 1994 年に松本によって考案され，1996 年に発表された[2]ものである． MBCF を実装するに当たっては，特殊なハードウェアを必要としない．通信のための API(Application Program Interface)が共有メモリ操作となっている点が，socket に代表されるこれまでの通信 API とは大幅に異なる点である．OS が提供する通信機構自体が共有メモリ操作を基本とする API を直接的に提供することにより，高機能，低遅延，低オーバヘッド，ロックフリーの通信同期機構の実現が可能になる．

MBCF は非常に多面的な特徴を持つため，概要を説明するだけで多くの紙面が割かれてしまう．より詳細な概要に関しては文献[1]を参照されたい．

3. 測定方法

3.1 性能測定内容

MBCF/Linux はマシン境界を越えた高機能な分散共有メモリ操作機能を提供する通信同期機構であるため，性能測定項目はいくらでも候補を考えることが可能である．しかし，複雑な共有メモリ操作の効果を測定するためには，該当する操作が効果を発揮できる状況を考案して，実験可能な状況を測定環境に創り出す必要がある．また，MBCF/Linux は完成して間もないため, MBCF/Linux を使用する並列分散アプリケーションも整備されていない．本稿

1 奈良女子大学
Nara Women's University
2 三菱電機株式会社
Mitsubishi Electric Corp.

Vol.2022-ARC-248 No.49
Vol.2022-SLDM-198 No.49
Vol.2022-EMB-59 No.49
2022/3/11

では，MBCF/Linux にとっての最初の性能評価として，他の通信機構とも簡単に対比可能なスループットとレイテンシに関して遠隔書き込み（MBCF_WRITE）コマンドを使用して測定を行った．ただし，MBCF/Linux は単純なデータ転送のみを目的とした通信機構ではないことには注意しておいて欲しい．

3.2 使用マシンの詳細

MBCF/Linux を用いて，AArch32(ARM の 32bit CPU)[6]のテストベッド（評価ボード）である TWR-LS1021A-PB[9]と AArch64(ARM の 64bit CPU)[7]のテストベッドである Raspberry Pi 4 model B (SRAM 4GB)[10]を用いた性能測定と x86_64(intel64[8] CPU)を搭載した PC を使って GbE（二種），10GbE（二種），100GbE(二種)の各 Ethernet NIC を使った基本性能のうちのスループットとラウンドトリップレイテンシを測定した．まず ARM のマシンについての詳細を以下に示す．

➢ ARM の 32bit CPU テスト環境

型番：NXP ® TWR-LS1021A-PB (または PC)
CPU：dual ARM Cortex-A7
メモリ：1GB DDR3L
Ethernet：TWR-LS1021A の GbE (物理層：AR8033×３)
OS：Linux 4.1.35-rt41 armv7l

➢ ARM の 64bit CPU テスト環境

型番：Raspberry Pi 4 model B
CPU：Cortex-A72
メモリ：4GB DDR4-2400
Ethernet：GbE BCM54213PE
OS：Linux ubuntu 4.19.80-v8+ aarch64

GbE（二種），10GbE（二種），100GbE(二種)の計測に使用したマシン（PC）は三種類で，各マシンの NIC，チップセット，マザーボード，CPU，メモリの詳細を以下に示す．

(1) マザーボード：Supermicro X11SCL-IF[17]
チップセット：intel® C242
CPU：intel(R) Core(TM) i3-9300T CPU @ 3.20GHz
メモリ：CFD W4U2666PS-8GC19×2

(2) マザーボード：Supermicro X11SCL-IF
チップセット：intel® C242
CPU：intel(R) Core(TM) i3-9100F CPU @ 3.60GHz
メモリ：CFD W4U2666PS-8GC19×2

(3) マザーボード：Supermicro X11SPM-F[18]
チップセット：intel® C621
CPU：intel(R) Xeon(R) Silver 4210R CPU @ 2.40GHz
メモリ：Kingston 9965690-055.A00G × 6

（１）と（２）のマシンの違いについては CPU の型番のみで，最大メモリーチャンネル数は 2，PCI Express レーンの最大数 は 16, CPU の対応ソケットは FCLGA1151 であるのに対して（３）のマシンについては，最大メモリーチャンネル数は 6，PCI Express レーンの最大数 は 48，CPU の対応ソケットは FCLGA3647 であり，メモリバンド幅と I/O バンド幅性能が大きく優れるマシンとなっている．

3.3 測定条件

今回の測定には性能測定用のテストプログラムを使い，2 台のマシン間で通信を行った．測定項目は，遠隔書き込みコマンドのみを対象として，ARM の 32bit，64bit のテストベッド上の GbE はパケット送信回数 10000 回，intel CPU マシンの GbE（二種），10GbE（二種），100GbE（二種）は送信回数 1000000 回，明示的確認応答間隔（何パケット毎に確認応答をするかという回数）0 回，1 パケットの書き込みデータサイズ 200~1440byte という条件とし，5 回分の測定結果を平均したデータを使用している．ラウンドトリップレイテンシは，パケット送信回数 60000 回，明示的確認応答間隔 1 回，往路パケットのデータサイズ 4byte と 1440byte の場合を計測した．ラウンドトリップレイテンシの復路は status 返り値(4byte)の返送を用い，元のパケットを送信し他ユーザプログラムが status 返り値の到着に気がつくまでの時間を測定した．測定データはスループットと同様に 5 回分の平均のデータである. intel CPU に関しては，ターボ・ブースト・テクノロジー等による CPU 動作周波数変動の影響を除去するために，実際の測定前に MBCF パケットを送りウォーミングアップとして CPU を動かしてからパケットを送っている．このため，スループットに関しては測定前に 80000 回，ラウンドトリップレイテンシに関しては 50000 回 MBCF パケットを送っている．

4. 測定結果

まず AArch32(ARM の 32bit)のテストベッド（評価ボード）である TWR-LS1021A-PB,PC を使った性能測定の結果について以下に示す．

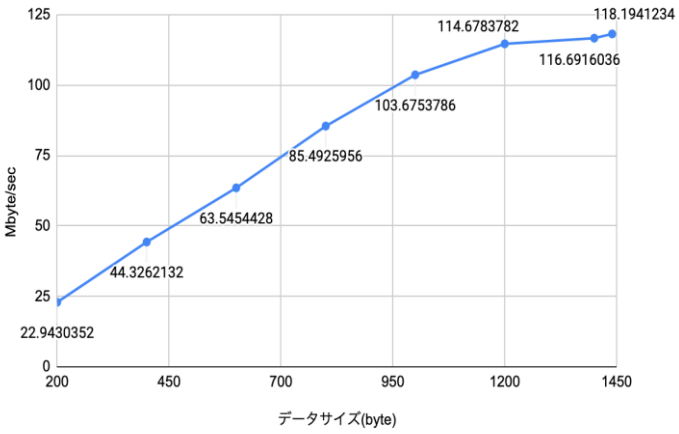

図 1 TWR-LS1021A(AArch32)での GbE 測定結果

Vol.2022-ARC-248 No.49
Vol.2022-SLDM-198 No.49
Vol.2022-EMB-59 No.49
2022/3/11

今回はスイッチングハブを通しての計測のみで，NETGEAR GS308V2 を使用した．図１よりデータサイズ 1440byte の時，約 117Mbyte/sec 以上の性能がでており，これはGbEの最大理論性能が125Mbyte/secであることとパケットのヘッダや Ethernet 自身のオーバヘッドを考慮すると，ほぼ理論性能であると言える．またラウンドトリップレイテンシは往路4byte復路4byteの時33.518μsec, 往路1440byte復路 4byte の時 59.976μsec であった．

次に AArch64(ARM の 64bit)のテストベッド（評価ボード）である Raspberry Pi 4 model B を使った性能測定の結果について以下に示す．

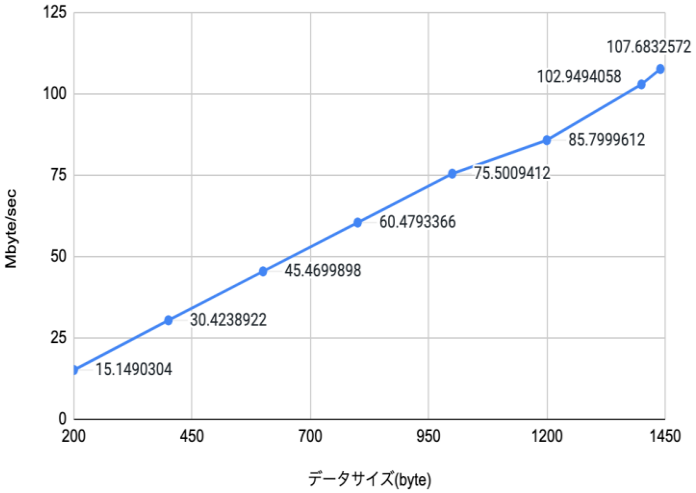

図２　Raspberry Pi 4(AArch64)での GbE 測定結果

AArch32 での計測で使用したスイッチングハブと同じものを使い計測した．図２より理論性能には達さない結果となった．この結果は実験環境のハードウェアもしくは NIC 用デバイスドライバに問題があると思われる．ラウンドトリップレイテンシは往路 4byte 復路 4byte の時 78.826μsec, 往路 1440byte 復路 4byte の時 122.424μsec となった．

次に（１）のマシンで intel GbE I210-AT[12]を使った性能測定の実験環境と結果について以下に示す．

マシン	（１）
NIC	intel GbE I210-AT
OS / Linux カーネル	Ubuntu18.04.5 LTS / 5.3.0-23-generic

表 1 GbE 実験環境（１）

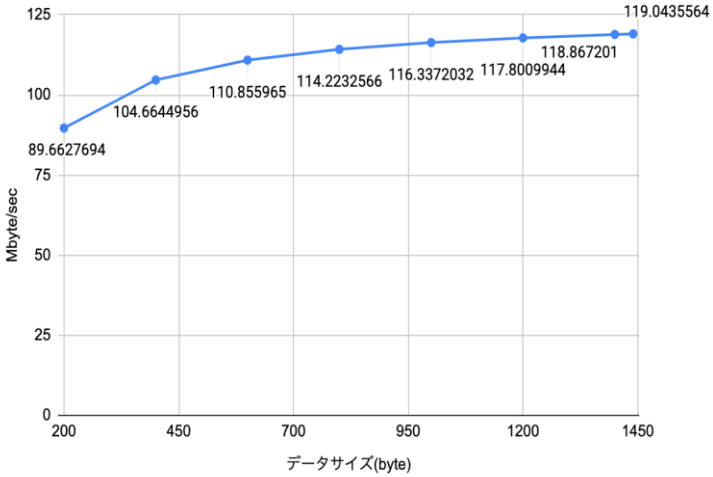

図３　GbE 実験環境（１）の測定結果

図 3 より，データサイズ 1200byte 以降からほぼ理論性能に達する性能が出ている．そしてラウンドトリップレイテンシは往路 4byte 復路 4byte の時 142.42μsec，往路 1440byte 復路 4byte の時 190.552μsec となった．また同様の環境で OS が Ubuntu20.04.1 LTS では，スループットはほとんど同じ結果で理論性能に達する結果となり，ラウンドトリップレイテンシは往路 4byte 復路 4byte の時 104.394μsec，往路 1440byte 復路 4byte の時 206.398μsec となった．

次に（３）のマシンで Intel GbE X722[11]を使った性能測定の実験環境と結果について以下に示す．

マシン	（３）
NIC	Intel GbE X722
OS / Linux カーネル	Ubuntu18.04.5 LTS / 5.3.0-23-generic

表２　GbE 実験環境（２）

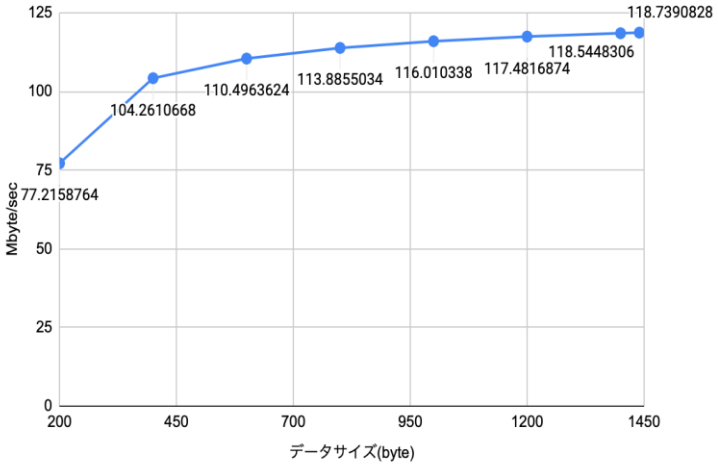

図４　GbE 実験環境（２）の測定結果

Vol.2022-ARC-248 No.49
Vol.2022-SLDM-198 No.49
Vol.2022-EMB-59 No.49
2022/3/11

図４より，マシン（３）でもマシン（１）と同様にほぼ理論性能が出ている.そしてラウンドトリップレイテンシは，往路 4byte 復路 4byte の時 72.19μsec，往路 1440byte 復路 4byte の時 126.39μsec となった.

次に（１）のマシンで intel 10GbE X540-T2[13]を使った性能測定の実験環境と結果について以下に示す.

マシン	（１）
NIC	intel 10GbE X540-T2
OS / Linux カーネル	Ubuntu18.04.5 LTS / 5.3.0-23-generic

表３　10GbE 実験環境（１）

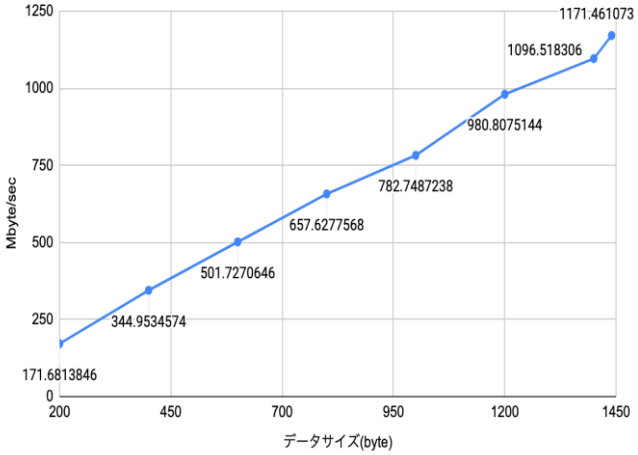

図５　10GbE 実験環境（１）の測定結果

図５より，10GbE でもデータサイズ 1440byte の時，GbE の理論性能である 117Mbyte/sec の 10 倍である約 1170Mbyte/sec を出しており理論性能に達していると言える．そしてラウンドトリップレイテンシは，往路 4byte 復路 4byte の時 34.772μsec，往路 1440byte 復路 4byte の時 61.262μsec となった.また同様の環境で OS が Ubuntu20.04.1 LTS では，スループットはほとんど同じ結果で理論性能に達する結果となり，ラウンドトリップレイテンシは往路 4byte 復路 4byte の時 30.864μsec，往路 1440byte 復路 4byte の時 61.312μsec となった.

次に（２）のマシンで Marvell 社の AQN107 10GbE[14]を使った性能測定の実験環境と結果について以下に示す.

マシン	（２）
NIC	Marvell AQN107
OS / Linux カーネル	Ubuntu18.04.5 LTS / 4.15.0-163-generic

表４　10GbE 実験環境（２）

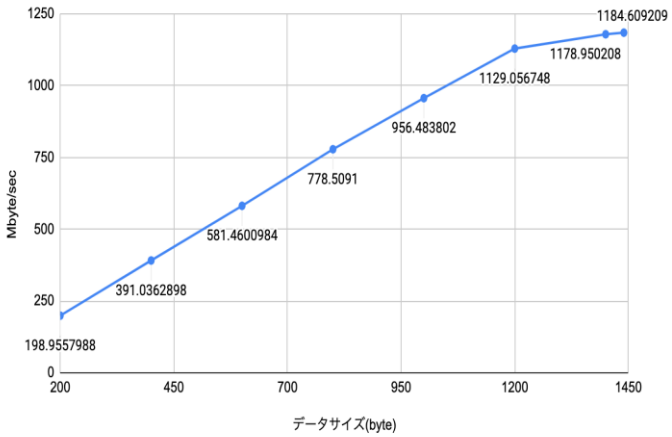

図６　10GbE 実験環境（２）の測定結果

図６より，Marvell 社の 10GbE でもデータサイズ 1400byte 以降から約 1170Mbyte/sec でており理論性能に達していると言える.そしてラウンドトリップレイテンシは往路 4byte 復路 4byte の時 47.638μsec，往路 1440byte 復路 4byte の時 53.76μsec となった.

次に（１）のマシンで intel 100GbE E810-CQDA1 [15]を使った性能測定の実験環境と結果について以下に示す.

マシン	（１）
NIC	intel 100GbE E810-CQDA1
OS / Linux カーネル	Ubuntu20.04.1 LTS / 5.4.0-42-generic

表５　100GbE 実験環境（１）

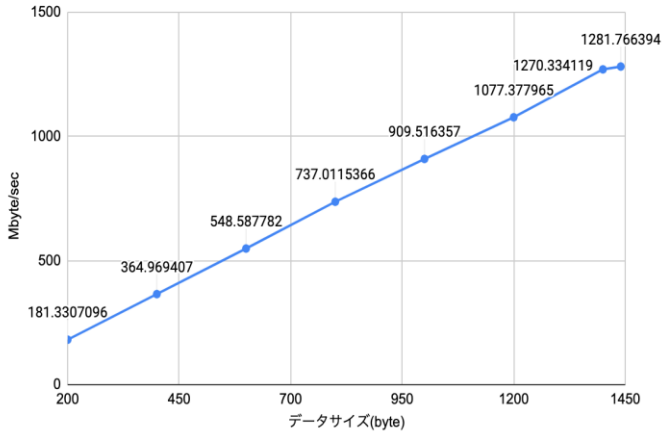

図７　100GbE 実験環境（１）の測定結果

図７より，mtu=1500 時の 100GbE のスループットは，理論性能に遠く及ばないというよりも，10GbE のスループットと大差ない結果となった．ラウンドトリップレイテンシは往路 4byte 復路 4byte の時 13.52μsec，往路 1440byte 復路 4byte の時 15.10μsec であった.

また同様の環境で mtu(maximum transfer unit)を 9000byte にした場合の測定も行ない，パケットのデータサイズ

Vol.2022-ARC-248 No.49
Vol.2022-SLDM-198 No.49
Vol.2022-EMB-59 No.49
2022/3/11

200~8920byte という条件で測定した結果を以下に示す．

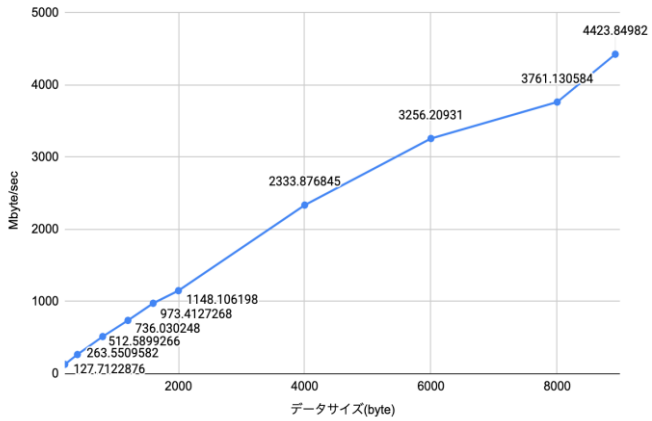

図８　100GbE 実験環境（１）での測定結果(mtu 9000)

次に（１）のマシンで NVIDIA Mellanox ConnectX-5 EN 515A-CCAT[16]を使った性能測定の実験環境と結果について以下に示す．

マシン	（１）
NIC	NVIDIA　Mellanox ConnectX-5 MCX 515A-CCAT
OS / Linux カーネル	Ubuntu20.04.1 LTS / 5.4.0-42-generic

表６　100GbE 実験環境（２）

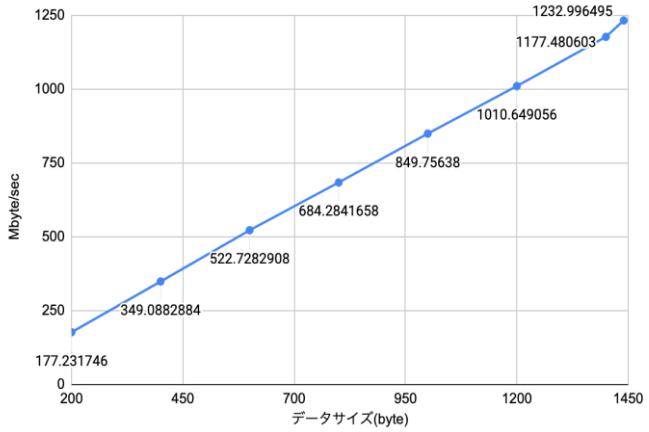

図９　100GbE 実験環境（２）の測定結果

図 9 より，mtu=1500 時の 100GbE のスループットは，理論性能に遠く及ばず，10GbE のスループットを少しだけ上回る結果となった．ラウンドトリップレイテンシは非常に優秀で，往路 4byte 復路 4byte の時 8.414μsec，往路 1440byte 復路 4byte の時 8.686μsec であった．

また同様の環境で mtu を 3680byte にした場合の測定も行ない，パケットのデータサイズ 200~3600byte という条件で測定した結果を以下に示す．なお，この NIC は HW もしくはデバイスドライバの不具合により，mtu=9000 では誤動作するため，mtu の値を正常に動作する上限値である 3680 にして測定を行った．

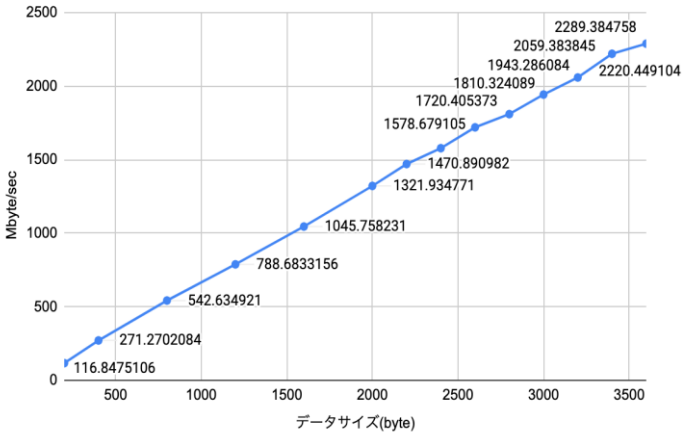

図１０　100GbE 実験環境（２）での測定結果(mtu 3680)

5. 結果の考察

データサイズ往路 4byte 復路 4byte の時のラウンドトリップレイテンシの値を以下の表７にまとめる．

Ethernet の実験環境	**μsec**
AArch32 TWR-LS1021A(Linux 4.1.35-rt41 armv7l)	33.518
AArch64 Raspberry Pi 4 (Linux ubuntu 4.19.80-v8)	78.826
intel GbE I210-AT（Ubuntu18.04.5 LTS）	142.724
intel GbE I210-AT（Ubuntu20.04.1 LTS）	104.394
intel GbE X722（Ubuntu18.04.5 LTS）	72.19
intel 10GbE X540-T2（Ubuntu18.04.5 LTS）	34.772
intel 10GbE X540-T2（Ubuntu20.04.1 LTS）	30.864
Marvell 10GbE AQN107（Ubuntu18.04.5 LTS）	47.638
intel 100GbE E810-CQDA1（Ubuntu20.04.1 LTS）	13.524
NVIDIA(Mellanox) ConnectX-5 MCX 515A-CCAT（Ubuntu20.04.1 LTS）	8.414

表７　ラウンドトリップレイテンシ（往復 4byte）

ARM CPU のテストベッドの二つはスイッチングハブを経由した時間であるため，数 μsec の誤差があると思われる．スイッチングハブの遅延を求めるために，intel GbE X722 の測定において，スイッチングハブを経由した場合と，2 台のマシンの NIC を直結した場合の測定を行ったが，有意な測定差は得られなかった．Linux OS の下では 64byte パケットがスイッチングハブを通過するレベルの時間は何か別の要因で隠蔽されるようである．ただし，そう考えると Mellanox の 100GbE のラウンドトリップレイテンシが約 8.4μsec と超高速であることの説明がつかない．ラウンドトリップレイテンシの値は NIC のハードウェアとデバイスドライバに大きく影響され，OS のバージョンや種類にもか

Vol.2022-ARC-248 No.49
Vol.2022-SLDM-198 No.49
Vol.2022-EMB-59 No.49
2022/3/11

なり影響されると言えそうである．Mellanox の 100GbE のラウンドトリップレイテンシの測定結果から MBCF/Linux は超低レイテンシのユーザレベル通信機構であると言うことが可能である．レイテンシの値があまり良くない実験環境については，100GbE のレイテンシから，その要因が MBCF/Linux デバイスドライバではないことは明らかであるが，引き続き値を悪くする外部要因(Linux や NIC HW や NIC ドライバ)を探って，可能であれば改善する方法を見つけたいと考えている．

スループットの値に関しては, GbE および 10GbE に関しては, AArch64 のテストベッドである Raspberry Pi 4 を除いて，Ethernet の標準 mtu である 1500byte で理論性能の上限を達成できている. 明らかに Raspberry Pi 4 には何らかの問題があって，GbE にも関わらずスループットの理論性能値が達成できないものと思われる．また，GbE に関して intel の GbE は非常に優秀でパケット内のデータペイロードが少ない時には，短い間隔でパケットの連続送信が可能なようである．このため，パケット内のデータサイズが小さい時でも ARM のテストベッドの GbE と比較してかなり高性能が達成されている．しかし，10GbE を見ると二種類のスループットのグラフの間に大きな差はない. 強いて言えば，Marvell AQN107 の方が若干スループット性能は上のようである．intel GbE のようなスループットのグラフが 10GbE では得られていないことには，送信オーバヘッドが 10GbE のスループットの理論性能上限を出すのにぎりぎりの値であるということが関連している．MBCF/Linux のデバイスドライバを mtu=9000byte で生成して（つまり sk_buff を 9000byte 確保して), Ethernet NIC は mtu=1500byte のままでスループットの測定を行うと，パケット内のデータサイズ 1440byte では 900Mbyte/sec ぐらいしか出ないことが判明している. 9000byte の sk_buff はページサイズを超えるため，Linux のパケット送信オーバヘッドが増えるからだと考えられる．このことは 100GbE のスループットの測定結果からも裏付けられる. mtu=1500byte では 10GbE と同程度のスループットしか達成できていない．つまり，パケット送信のオーバヘッドがボトルネックとなって 100GbE を使い切ることができなくなっている．mtu の値を 9000byre にしても約 4400Mbyte/sec が上限である. これは 100GbE の転送能力の 35%程度しか使い切れていない．もっと詳しくボトルネックの原因を探るために, sk_buff 層のパケット送信関数である dev_queue_xmit()を始めとする MBCF/Linux の送信関数に含まれる Linux の提供関数の実行時間（オーバヘッドコスト）を測定したところ，CPU や NIC によってばらつきはあるものの 0.5μsec から 1.0μsec の送信コストが掛かっていることが判明した．このことから，他のコストが 0 であっても，現状の MBCF/Linux はパケットを一つずつ送信しているため，1 秒間にせいぜい 100 万から 200 万のパケットしか送信することができない. このことは, mtu が 1500 であれば，1.5Gbyte/sec から 3Gbyte/sec が上限であることを意味する．TCP/IP では，大量のデータ転送時に IP のセグメントサイズを大きくして, 1 回の dev_queue_xmit()の呼び出しで複数個（最大 16 個）のパケットを送信しているものと思われる．100GbE におけるスループットについて，送信オーバヘッドコストによるボトルネックを打ち破って理論性能に近いスループットを達成するためには，MBCF/Linux においても同様に 1 回の送信関数の呼び出しで, sk_buff 層においても複数個のパケットをまとめて飛ばす送信方法を導入する必要があると思われる．

Mellanox 100GbE の mtu=1500byte の最大スループットは MBCF/Linux が動き始めた当初（2021 年 1 月頃）は 600Mbyte/sec 程度しか出なかった．これは，明示的もしくは暗示的な確認応答が戻る前に先出し可能なパケット数の上限が少なすぎたためである．この事実が判明して，先出し可能なパケット数を 100 倍程度に大きくしてチューニングを行った．その結果，100GbE に関しては今回報告した値とほとんど同じ性能になったが, 逆に GbE のスループットが大幅に悪化した．MBCF は受信側でバッファリングしておらず到着順序を守るため，先出しするパケット数が多過ぎてパケット喪失が発生すると大きなペナルティを被ることになる．このことから，先出しパケット数の上限は Ethernet の回線スピードによって調整するようにデバイスドライバのコードを改良した．今回の測定結果は，この改良を終えた MBCF64/Linux デバイスドライバによるものであり，測定に使われたすべてのデバイスドライバは同一のソースコードから生成されている．

6. おわりに

オリジナル並列分散 OS である SSS-PC および SSS-CORE の基盤となる通信同期機構として採用していた Memory Based Communication Facility (**MBCF**)の Linux OS 版である MBCF/Linux を開発するとともに, MBCF を 64bit アドレスに対応させた MBCF64 を新たに規定して，64bit アドレスの Linux OS にも対応可能にした．本稿では，MBCF/Linux のスループットとレイテンシの基本性能測定結果について報告した．ユーザレベルのプログラムによる性能測定において，100GbE のラウンドトリップレイテンシで約 8.4μsec という超低レイテンシを達成し，スループットにおいても GbE と 10GbE の NIC であれば二つのユーザプログラム間で理論性能上限のデータ転送能力を達成できた．100GbE NIC のスループットに関しては，Linux のパケット送信部やシステムコール呼び出しがボトルネックになって，理論性能上限の性能を達成することはできなかった．MBCF/Linux が単体パケットによる送受信を行っていて，パケット間においてストロングコンシステンシを維持していることを考慮すれば，達成されたスループットは十分に高い値だと考えている．ただし，将来的には TCP/IP

Vol.2022-ARC-248 No.49
Vol.2022-SLDM-198 No.49
Vol.2022-EMB-59 No.49
2022/3/11

にスループットの項目においても負けないようにしたいと考えている．そのために，複数パケットを 1 回の MBCF/Linux の送信呼び出しで送信可能なインタフェースを作成し，送信処理内部の sk_buff 層も複数パケットをまとめて送信する方法を Linux ソースコードから解読して実装し，ぜひ TCP/IP に負けないスループットを MBCF/Linux において達成したいと考えている．

謝辞 ARM AArch32 を対象にした MBCF32/Linux の開発と測定は奈良女子大学と三菱電機株式会社先端技術総合研究所との共同研究「組込み SoC クラスタ化技術に関する研究」の下で実施された．関係者のみなさまに深く感謝申し上げます．また，日頃の研究活動と大学生活を支援してくれている松本研究室の所属メンバに心より感謝いたします．

参考文献

[1] 松本 尚: Linux 版の MBCF 通信機構について．コンピュータシステムシンポジウム論文集, 2021，情報処理学会,pp.27--36 (December 2021).
[2] 松本 尚，他: 汎用超並列オペレーティングシステム: SSS-CORE --- ワークステーションクラスタにおける実現 ---．システムソフトウェアとオペレーティングシステム研究会報告 No.73-20，情報処理学会, pp.115--120 (August 1996).
[3] 松本 尚: 次世代オペレーティングシステム SSS-PC の開発 --- IA32 用カーネルアーキテクチャ ---. ITX2002 Summer 論文集（CDROM），情報処理振興事業協会, (June 2002)
[4] Matsumoto, T. et al.: MBCF: A Protected and Virtualized High-Speed User-Level Memory-Based Communication Facility. Proc. of the 1998 ACM Int. Conf. on Supercomputing, pp.259--266 (July 1998).
[5] Matsumoto, T.: A Study on Memory-Based Communications and Synchronization in Distributed-Memory Systems. Dissertation Thesis, Graduate School of Science, Univ. of Tokyo (February 2001).
[6] Arm Limited: ARM Architecture Reference Manual Supplement - ARMv8, for the ARMv8-R AArch32 architecture profile. https://developer.arm.com/documentation/ddi0568/latest (2022 年 1 月 9 日確認).
[7] Arm Limited: Arm Architecture Reference Manual Supplement - Armv8, for Armv8-R AArch64 architecture profile. https://developer.arm.com/documentation/ddi0600/ac (2022 年 1 月 9 日確認).
[8] intel: Intel® 64 and IA-32 Architectures Software Developer Manuals.https://www.intel.com/content/www/us/en/developer/articles/technical/intel-sdm.html (2022 年 1 月 9 日確認).
[9] NXP: QorIQ LS1 Tower System Module - Fact sheet https://www.mouser.jp/datasheet/2/302/TWRLS1021AFS-793220.pdf (2022 年 1 月 5 日確認).
[10] Raspberry Pi Foundation: raspberry-pi-4-product-brief.pdf https://datasheets.raspberrypi.com/rpi4/raspberry-pi-4-product-brief.pdf (2022 年 1 月 5 日確認).
[11] intel: Intel® Ethernet Network Adapter X722 https://www.intel.co.jp/content/www/jp/ja/ethernet-products/network-adapters/ethernet-x722-brief.html (2022 年 1 月 5 日確認).
[12] intel: Intel® Ethernet Controller I210 https://ark.intel.com/content/www/jp/ja/ark/products/64400/intel-ethernet-controller-i210at.html (2022 年 1 月 5 日確認).
[13] intel: Intel Ethernet Converged Network Adapter X540-T2 https://ark.intel.com/content/www/jp/ja/ark/products/60020/intel-ethernet-controller-x540at2.html (2022 年 1 月 5 日確認).
[14] Marvell: Marvell® AQtion AQN-107 10 GbE Network Interface Adapter https://www.marvell.com/content/dam/marvell/en/public-collateral/ethernet-adaptersandcontrollers/marvell-ethernet-adapters-aqtion-aqn107-product-brief.pdf (2022 年 1 月 5 日確認).
[15] intel: インテル® イーサネット・ネットワーク・アダプター E810-CQDA1 https://www.intel.co.jp/content/www/jp/ja/products/sku/192561/intel-ethernet-network-adapter-e810cqda1/specifications.html (2022 年 1 月 5 日確認).
[16] NVIDIA: ConnectX® -5 EN Card https://www.mellanox.com/files/doc-2020/pb-connectx-5-en-card.pdf (2022 年 1 月 5 日確認).
[17] Supermicro: X11SCL-iF https://www.supermicro.com/ja/products/motherboard/X11SCL-IF (2022 年 1 月 5 日確認).
[18] Supermicro: X11SPM-F https://www.supermicro.com/ja/products/motherboard/X11SPM-F (2022 年 1 月 5 日確認).