

# スループットを用いた「京」における MPI 通信性能の評価

北澤好人<sup>†1</sup> 黒田明義<sup>†1</sup> 志田直之<sup>†2</sup> 安達知也<sup>†2</sup> 南一生<sup>†1</sup>

**概要:** スーパーコンピュータ「京」は 82,944 ノードから構成される超並列のシステムであり、アプリケーションにて高い実効性能を得るためには、MPI 通信関数の性能が重要である。本稿では、スーパーコンピュータ「京」における MPI 通信性能を測定し、実測したスループットのピーク性能と半性能長から、通信バンド幅とレイテンシを算出した。また通信モデルから通信時間の見積りを理論的に導出し、測定値と比較することで性能の評価を行った。その結果、隣接通信ではピーク性能と通信バンド幅がほぼ一致して、遅延時間は通信のレイテンシによる値と概ね合っていることがわかった。集団通信では Bcast, Allreduce, Allgather について同様の評価を行い、ピーク性能が約 10%~20% 低く、遅延時間は約 10%~50% の違いがあるものの、測定結果が通信時間見積りと同様の傾向であることがわかった。

**キーワード:** スーパーコンピュータ「京」、MPI 通信性能評価、Intel MPI ベンチマーク、通信バンド幅、レイテンシ、

## Evaluation of MPI communication performance using throughput on the K computer

YOSHITO KITAZAWA<sup>†1</sup> AKIYOSHI KURODA<sup>†1</sup> NAOYUKI SHIDA<sup>†2</sup>  
TOMOYA ADACHI<sup>†2</sup> KAZUO MINAMI<sup>†1</sup>

**Abstract:** The K computer is a massively parallel system with 82,944 nodes, and the performance of the MPI communication function is extremely important to obtain high effective performance for the application. In this paper, we measured the MPI communication performance on the K computer and calculated the communication bandwidth and latency from the peak performance and the half-performance length of measured throughput. Furthermore, we derived the estimation formula of the communication time theoretically from the communication model, and evaluated the communication performance by comparing with the measured value. As a result, in nearest neighbor communication, the peak performance almost matched the communication bandwidth, and the delay time roughly matched the communication latency. In collective communication, we evaluated Bcast, Allreduce and Allgather similarly. The peak performance was about 10%-20% lower and delay time was about 10%-50% different, but measurement results were the same trend as estimation results.

**Keywords:** K computer, performance of MPI communication, Intel MPI Benchmarks, band width, latency

## 1. はじめに

スーパーコンピュータ「京」(以下、「京」と記す)では、様々な分野のアプリケーションプログラムが実行されている。並列規模は数千~数万であり、並列化のための通信処理として MPI 通信が必須である。

本稿では、Intel MPI Benchmarks[1] (以下、IMB と記す)を用いてスーパーコンピュータ「京」における主要な MPI 通信の性能測定を行い[2]、実測されたスループットのピーク性能と半性能長から、通信バンド幅とレイテンシを算出した。また通信モデルから通信時間の見積りを理論的に導出し、測定値との比較を行ったのでその結果を報告する。2 章では「京」の通信の概要、3 章ではスループットの概要を説明し、4 章にて通信性能を検証した内容を報告する。5 章では展開と今後の課題をまとめる。

## 2. 「京」の通信性能の概要

本章では、「京」における計算ノード間の通信機構の概

要を説明する。

### 2.1 通信性能の概要

「京」では、計算ノード間の通信ネットワークを構成するために、Tofu(Torus fusion)インターコネクト[3][4]が採用されている。4 つの Tofu ネットワークインタフェース(TNI)により、最大で 4 方向の同時通信が可能である。また Tofu ネットワークルータ(TNR)は 10 本のリンクを持ち、各リンクの理論上の通信バンド幅は 5[GB/s](双方向)である[3]。実際には MPI のオーバーヘッドにより、通信バンド幅の実効値は約 4.5[GB/s]である。ただし、同時通信についてはハードウェアのバスの性能制限により、1 ノードあたりの総通信バンド幅が約 15[GB/s]に律速される[5][6]。「京」ではこの計算ノードが 6 次元メッシュトラスの直接網で接続されており、ジョブ毎に 3 次元トラスを切り出せる。

### 2.2 MPI 環境

「京」の MPI 環境は Open MPI をベースに開発されている。MPI の集団通信関数については、Open MPI 由来の通信アルゴリズムの他に、Tofu インターコネクト向けに最適化

<sup>†1</sup> 理化学研究所  
<sup>†2</sup> 富士通株式会社

RIKEN  
FUJITSU,LTD.

された通信アルゴリズム (Tofu 専用アルゴリズム) を開発し、高い通信性能を実現している。例えば Bcast 通信では、3 次元ノード形状の場合、メッセージを 3 分割して 3 方向同時通信をする通信アルゴリズム Trinaryx3 を新規に開発した[5][7]。これは直接網ネットワークポロジや同時通信の機構を活用したものとなっている。Open MPI のアルゴリズムと Tofu 専用アルゴリズムは、通信時のメッセージサイズに応じて、最適なアルゴリズムが自動的に選択される。

### 3. 通信のスループットの概要

通信性能を評価する指標の 1 つとして、スループットがある。スループットとは単位時間あたりに転送するメッセージサイズで、メッセージサイズを通信時間で割った値である。通信では、転送の初期にメッセージが到着するまでに遅延時間  $t_L$  が発生し、その後、メッセージが転送時間  $t_M$  で順次転送される。転送するメッセージサイズを  $M$  とし、通信バンド幅を  $B$  として、全体の通信時間  $t$  は、以下のよう

$$t = t_L + t_M = t_L + \frac{M}{B} \quad (1)$$

スループット  $T_p$  は単位時間  $t$  あたりのメッセージサイズ  $M$  であるので、以下となる。

$$T_p = \frac{M}{t} = \frac{M}{t_L + M/B} = \frac{B}{1 + (Bt_L)/M} \quad (2)$$

$T_p = B/2$  となる通信量を  $M_{1/2}$  とした時、これを半性能長と呼ぶ。半性能長を使ってスループットを表すと、以下のようになる。

$$T_p = B / \{1 + 1/(M/M_{1/2})\} \quad (3)$$

$$M_{1/2} = Bt_L \quad (4)$$

これを、ベクトル演算の性能で用いられた R.W.Hockney の pipe 関数[8]を用い、スループットは以下のように書ける。

$$T_p = B \text{ pipe}(M/M_{1/2}) \quad (5)$$

$$\text{pipe}(x) \equiv 1/(1 + 1/x) \quad (6)$$

pipe 関数は、 $x=1$  の時に  $1/2$  となり、 $x$  が大きい極限で  $1$ 、 $0$  の極限で  $0$  である。式(5)からスループット  $T_p$  はメッセージサイズ  $M$  が大きい極限でピーク性能である通信バンド

幅  $B$  に近づき、メッセージサイズが小さい場合、遅延時間  $t_L$  の比率が大きくなる。半性能長  $M_{1/2}$  はピーク性能  $B$  と通信レイテンシ  $t_L$  がバランスする点であり、半性能長  $M_{1/2}$  から、通信レイテンシ  $t_L$  を見積もることが可能である。

$$B = \lim_{M \rightarrow \infty} (T_p) \quad (7)$$

$$t_L = M_{1/2}/B \quad (8)$$

### 4. スループットによる性能評価

通信のスループットを用いて、IMB の一対一通信の PingPong と集団通信の Bcast, Allreduce, Allgather の性能を評価した。測定に用いたベンチマークプログラムは IMB の Ver.3.2.4 で、通信ライブラリを含む「京」の言語環境のバージョンは、2016 年 5 月から 2017 年 1 月までの間、最新版として提供された K-1.2.0-20-1 である。

測定結果に対して、前述のスループットの式(3)を最小二乗法によりフィッティングして、通信性能の指標となるピーク性能  $B$  と半性能長  $M_{1/2}$  (または遅延時間  $t_L$ ) を算出した。また、通信モデルから通信時間の見積式を理論的に導出し、そこから算出した通信性能を比較し、評価を行った。この評価式は、システム設計時の通信性能予測や実際の完成したシステム性能評価を行うことを目的に、「京」開発時に通信性能を評価するために導出されたモデル式を、Tofu 専用アルゴリズムを考慮して拡張したものである。今回の評価で得られた各通信方式の通信性能を表 1 にまとめた。上段が実測値から得られた値で、下段が後述する見積式から算出した値である。

#### 4.1 一対一通信の性能評価 (PingPong)

本節では、一対一通信として IMB の PingPong の性能を説明する。PingPong は、2 つのランク間において、一方のランクからデータを Send 通信により送信し、もう一方のランクにて受信した後に Recv 通信によりデータを元のランクへ送り返すことで、データを往復させる。

##### (1) 測定結果のフィッティング

「京」の MPI による一対一通信は 13[KiB]から通信アルゴリズムが変わる[5]ため、測定結果の 13[KiB]以上のデー

表 1 通信時間の実測値並びに見積式から得られた通信性能

Table 1 Communication performance parameter by measurement and estimation

|            | Parameter        | PingPong  | Bcast<br>1D/16KiB | Bcast<br>2D/16KiB | Bcast<br>3D/16KiB | Allreduce<br>3D/16KiB | Allgather<br>3D/~32KiB | Allgather<br>3D/64KiB~ |
|------------|------------------|-----------|-------------------|-------------------|-------------------|-----------------------|------------------------|------------------------|
| Fitting    | $B$ [MB/s]       | 4,685     | 3,616             | 7,467             | 10,400            | 5,266                 | 149,800                | 13,210                 |
|            | $M_{1/2}$ [byte] | 4.894E+04 | 1.010E+07         | 4.092E+06         | 1.756E+06         | 1.024E+06             | 6.206E+05              | 6.917E+03              |
|            | $t_L$ [ $\mu$ s] | 10.45     | 2,795             | 548.0             | 168.9             | 388.8                 | 1,590                  | 201.1                  |
| Estimation | $B$ [MB/s]       | 4,500     | 4,500             | 9,000             | 13,500            | 6,306                 | 1,920,000              | 15,040                 |
|            | $M_{1/2}$ [byte] | 3.134E+04 | 9.078E+06         | 3.344E+06         | 1.550E+06         | 1.350E+06             | 7.669E+07              | 9.470E+03              |
|            | $t_L$ [ $\mu$ s] | 6.965     | 2,017             | 371.6             | 114.8             | 428.2                 | 1,534                  | 241.8                  |

タを用いて、最小二乗法によるフィッティングを行った。

求めた値は表 1 の通りである。ピーク性能  $B$  は、4,685[MB/s]で通信バンド幅の実効性能限界に近い値である。スループットの測定結果とフィッティングの比較を図 1 に示す。丸印がスループットの測定結果で実線がフィッティング結果である。破線の範囲はフィッティングを行ったメッセージサイズである。

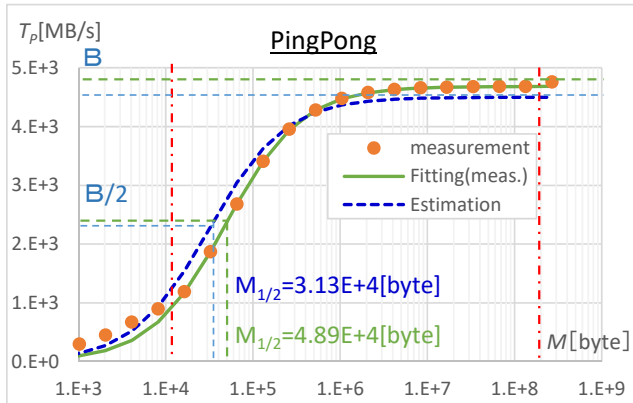


図 1 通信時間の測定結果と見積式から算出したスループットの比較(PingPong)

Figure 1 Comparison of throughput calculated from measurement and estimation (PingPong)

## (2) 見積式からの通信時間の算出

通信時間はメッセージの転送時間と遅延時間の合計であり式(1)で記述できる。遅延時間  $t_L$  には、MPI レイテンシと一対一通信のための制御通信の時間が含まれる。MPI レイテンシは、「京」において隣接ノード間の片側転送の時間として、1.6 [μs]が実測されている。13[KiB]以上の一対一通信では、データ転送の前に格納先のアドレス等の情報を転送する制御通信が行われる[5]。「京」での制御通信は、3種類の情報 (40[byte], 64[byte], 80[byte]) が通信前後に転送される。それぞれ制御通信を含まない 13[KiB]以下の PingPong の通信時間を用いて評価すると表 2 となる。

従って、遅延時間は合計で 6.965 [μs]となる。ピーク性能  $B$  は通信バンド幅の実効値を用いた。これらの数字を式(4)に代入すると、半性能長  $M_{1/2}$  が算出できる (表 1)。通信時間の見積式と測定結果のフィッティングによるスループ

表 2 制御通信時間

Table 2 Time of control communication

| Size      | Time [μs] |
|-----------|-----------|
| 40 [byte] | 1.751     |
| 64 [byte] | 1.829     |
| 80 [byte] | 1.785     |
| Total     | 5.365     |

ットの比較は図 1 の通りである。点線が見積式による結果である。見積式による遅延時間は約 7.0 [μs]であり、測定結果による遅延時間の約 10.5[μs]と比べて短い。遅延時間の内訳は、ある程度、制御通信時間と MPI レイテンシを用いて説明可能であることがわかる。

## 4.2 集団通信の性能評価

本節では、IMB の集団通信の Bcast, Allreduce, Allgather の通信性能を説明する。Bcast はルートランクから全ランクにデータを転送する。実行時のノード形状は 1 次元、2 次元、3 次元の 3 種類について通信性能を評価する。Allreduce は全ランクのデータを集めて集合演算を行い、その結果を全ランクに転送する。Allgather は全ランクが持つデータを集めて全ランクへ転送する。Allreduce と Allgather については、実行時のノード形状が 3 次元での通信性能を評価する。

### 4.2.1 Bcast (1 次元)

#### (1) 測定結果のフィッティング

384 ノード(1 次元ノード割当)における Bcast の測定結果について、PingPong と同様に評価した。「京」の Bcast 通信は Tofu 専用アルゴリズムの Trinaryx3 が適用され、パイプライン方式によりデータが転送される。パイプライン転送のイメージを図 2 に示す。パイプライン方式ではデータをセグメントに分割して転送を行い、実際の通信ではメッセージサイズによって最適なセグメントサイズが選択される。ここではセグメントサイズを 16[KiB]に固定して測定を行った。Tofu 専用アルゴリズムの Trinaryx3 は 128[KiB]以上のデータを用いて最小二乗法によるフィッティングを行った。表 1 に求められた通信性能を、図 3 にスループットの測定結果とフィッティング結果を示す。丸印の点が測定結果で実線がフィッティング結果である。

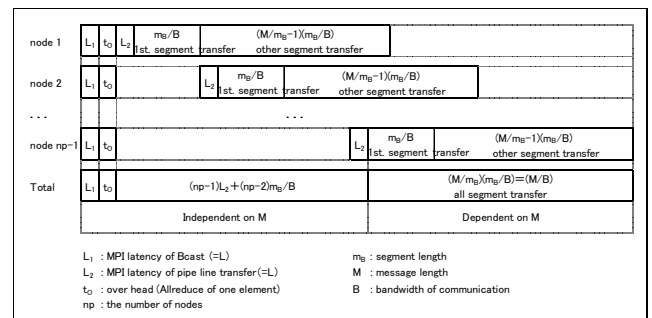


図 2 Bcast 通信のパイプライン転送のイメージ

Figure 2 Concept of pipeline transfer

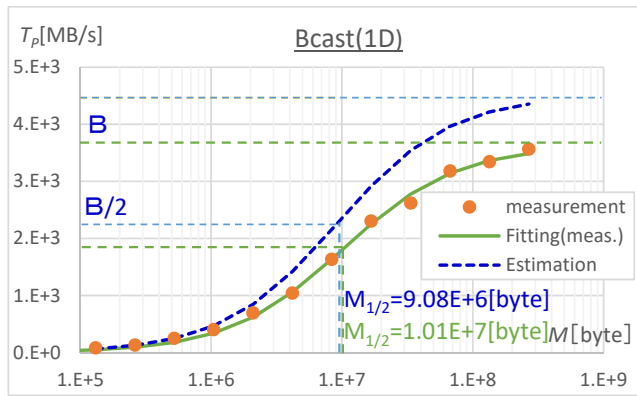


図 3 通信時間の測定結果と見積りから算出したスループットの比較 (Bcast/1D/16KiB)

Figure 3 Comparison of throughput calculated from measurement and estimation (Bcast/1D/16KiB)

## (2) 見積りからの通信時間の算出

Trinaryx3 はデータをセグメントに分割して、パイプライン方式により次のランクへ順次転送する。図 2 からパイプライン方式の通信時間は、メッセージサイズ  $M$  に依存しない時間と依存する時間の合計である。メッセージサイズ  $M$  に依存しない時間には、全体の MPI レイテンシ  $L$  とアルゴリズム判定処理のための 1 要素の allreduce などによるオーバーヘッド  $t_0$  に加え、パイプライン転送において、隣接ノード間当りに MPI レイテンシと 1 セグメントのデータの転送時間が含まれる。転送ノード総数を  $np$  とし、1 セグメントのサイズを  $m_B$  とおき、式(1)を書き下すと通信時間  $t$  は以下のように見積もられる。

$$t = (np - 1)(L + m_B/B) + L + t_0 + (M/m_B) \max(L, m_B/B) \quad (9)$$

メッセージサイズ  $M$  に依存しない項を抽出すると遅延時間  $t_L$  は、今回測定に用いたセグメントサイズの条件  $L < m_B/B$  にて、以下となる。

$$t_L = (np - 1)(L + m_B/B) + L + t_0 \quad (10)$$

評価に用いた各パラメータの値を表 3 に示す。通信時間の見積りから算出された通信性能は表 1 の通りである。 $t_0$  の値は、評価で使用したノード形状を用いて実測したものである。通信時間の見積りと測定のフィッティングによるスループットの比較は図 3 の通りである。点線が見積りに

表 3 見積りに用いた評価パラメータ (Bcast/1D/16KiB)

Table 3 Evaluation parameter (Bcast/1D/16KiB)

| Name  | Value            |
|-------|------------------|
| $np$  | 384 [node]       |
| $B$   | 4,500 [MB/s]     |
| $m_B$ | 16,384 [byte]    |
| $L$   | 1.600 [ $\mu$ s] |
| $t_0$ | 8.370 [ $\mu$ s] |

よる結果である。

ピーク性能  $B$  は、通信バンド幅の実効値 4,500[MB/s]に比べ、測定結果では 3,616[MB/s]と約 20%低い。遅延時間  $t_L$  は、通信時間の見積りから求めた遅延時間 2,017[ $\mu$ s]と比べて、測定結果での値は 2,795[ $\mu$ s]と約 39%の違いがあり、PingPong の遅延時間 10.5[ $\mu$ s]と比較すると大きく異なる。両者の通信時間の見積りを比べると、Bcast ではノード数に比例して増加するパイプラインの立ち上げ時間が含まれることが主要因であると考えられる。ノード数が増えることでパイプラインの立ち上げ時間が増加して、半性能長と遅延時間が増加する。

## 4.2.2 Bcast (2 次元)

### (1) 測定結果のフィッティング

64×6=384 ノード (2 次元ノード割当) における Bcast の測定結果について、1 次元と同様にセグメントサイズを 16[KiB]と固定して、128[KiB]以降のデータを用いて、最小二乗法によるフィッティングを行った。表 1 に求められた通信性能を、図 4 にスループットの測定結果とフィッティング結果を示した。丸印が測定結果で実線がフィッティング結果である。

2 次元ノード割当の場合、データを 2 分割して 2 方向に同時通信を行う。このため、1 次元に比べて、ピーク性能  $B$  は約 2 倍となる。

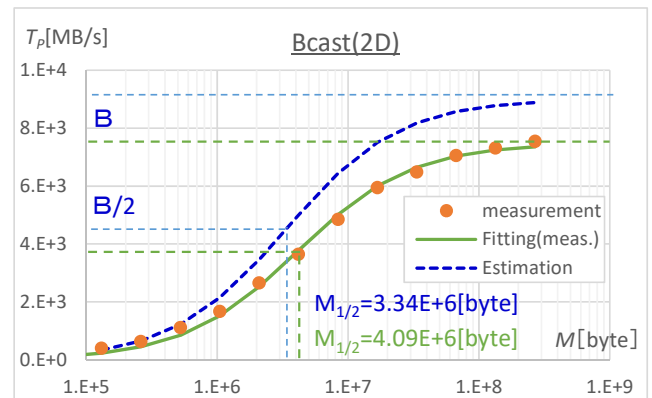


図 4 通信時間の測定結果と見積りから算出したスループットの比較 (Bcast/2D/16KiB)

Figure 4 Comparison of throughput calculated from measurement and estimation (Bcast/2D/16KiB).

## (2) 見積りからの通信時間の算出

2 次元ではノード割当が  $np1 \times np2$  ( $=np$ ) となり、通信時間の内訳は 1 次元と比べて、データを 2 分割して 2 方向に同時通信をするため、1 方向当りの通信では、 $M \rightarrow (M/2)$ 、 $B \rightarrow (B/2)$  となる。式(9)から通信時間の見積りを以下の通りである。

$$t = ((np1 - 1) + (np2 - 1) + 1)(L + m_B/(B/2)) + L + t_o + ((M/2)/m_B) \max(L, m_B/(B/2)) \quad (11)$$

メッセージサイズ  $M$  に依存しない項を抽出すると、遅延時間  $t_L$  は、今回測定に用いたセグメントサイズの条件  $L < m_B/(B/2)$  にて、以下となる。

$$t_L = (np1 + np2 - 1)(L + m_B/(B/2)) + L + t_o \quad (12)$$

尚、Trinaryx3 の場合、パイプラインのステップ数  $((np1-1) + (np2-1) + 1 = np1 + np2 - 1)$  が、1次元のステップ数  $(np1-1)$  に比べて小さいため、 $np1 > 1, np2 > 1$  の条件で、遅延時間  $t_L$  は1次元に比べて短くなる。

$$np1 + np2 - 1 \leq np1 \cdot np2 - 1 = np - 1 \quad (13)$$

評価に用いた各パラメータを表 4 に示す。通信時間の見積式から算出した通信性能は表 1 の通りである、通信時間の見積式と測定のフィッティングによるスループットの比較は図 4 の通りである。点線が見積式による結果である。

表 4 見積式に用いた評価パラメータ (Bcast/2D/16KiB)

Table 4 Evaluation parameter (Bcast/2D/16KiB)

| Name  | Value            |
|-------|------------------|
| $np1$ | 64 [node]        |
| $np2$ | 6 [node]         |
| $B$   | 9,000 [MB/s]     |
| $m_B$ | 16,384 [byte]    |
| $L$   | 1.600 [ $\mu$ s] |
| $t_o$ | 8.370 [ $\mu$ s] |

ピーク性能  $B$  は、通信バンド幅の 2 方向分の実効値 9,000[MB/s]に比べ、測定結果では 7,467[MB/s]と約 17%低く 1次元と同じ傾向である。遅延時間  $t_L$  は、通信時間の見積式から求めた値が 372[ $\mu$ s]と比べて、測定結果の値は 548[ $\mu$ s]と約 47%の違いがある。1次元の値と比較すると、パイプラインのステップ数が 383 から 69 へと小さくなるため、遅延時間も 1次元の 2,795[ $\mu$ s]から 548[ $\mu$ s]へ減少する。

#### 4.2.3 Bcast (3次元)

##### (1) 測定結果のフィッティング

$8 \times 6 \times 8 = 384$  ノード (3次元ノード割当)における Bcast の測定結果について、1次元と同様にセグメントサイズを 16[KiB]と固定して、128[KiB]以降のデータを用いて、最小二乗法によるフィッティングを行った。表 1 に求められた通信性能を、図 5 にスループットの測定結果とフィッティング結果を示した。丸印が測定結果で実線がフィッティング結果である。

3次元ノード割当の場合、データを 3 分割して 3 方向に同時通信を行うので、1次元に比べて、ピーク性能  $B$  は約 3 倍となる。

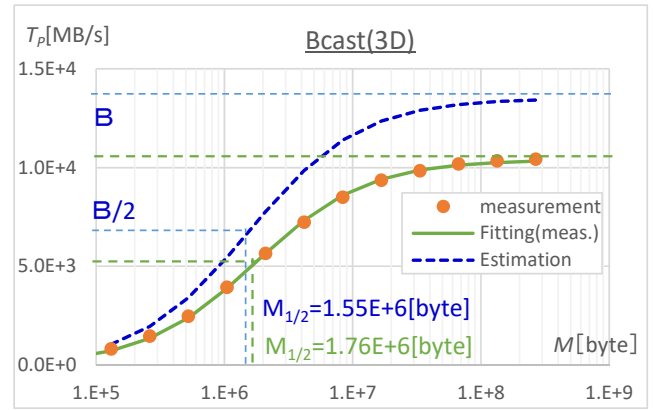


図 5 通信時間の測定結果と見積式から算出したスループットの比較 (Bcast/3D/16KiB)

Figure 5 Comparison of throughput calculated from measurement and estimation (Bcast/3D/16KiB)

##### (2) 見積式からの通信時間の算出

3次元ではノード割当が  $np1 \times np2 \times np3 (=np)$  となり、データを 3 分割して 3 方向に同時通信する。このため、通信時間の内訳は 1次元と比べて、1方向当りの通信では、 $M \rightarrow (M/3)$ ,  $B \rightarrow (B/3)$  となる。通信時間の見積式は式(9)から以下になる。

$$t = ((np1 - 1) + (np2 - 1) + (np3 - 1) + 1)$$

$$(L + m_B/(B/3)) + L + t_o + ((M/3)/m_B)$$

$$\max(L, m_B/(B/3)) \quad (14)$$

遅延時間  $t_L$  は、測定に用いたセグメント長の条件  $L < m_B/(B/3)$  では以下となる。

$$t_L = (np1 + np2 + np3 - 2)(L + m_B/(B/3))$$

$$+ L + t_o \quad (15)$$

尚、Trinaryx3 の場合、パイプラインのステップ数  $((np1-1) + (np2-1) + (np3-1) + 1 = np1 + np2 + np3 - 2)$  が、1次元のステップ数  $(np1-1)$  に比べて小さいため、 $np1 > 1, np2 > 1, np3 > 1$  の条件で、遅延時間  $t_L$  は1次元に比べて短くなる。

$$np1 + np2 + np3 - 2 \leq np1 \cdot np2 \cdot np3 - 1$$

$$= np - 1 \quad (16)$$

表 5 見積式に用いた評価パラメータ (Bcast/3D/16KiB)

Table 5 Evaluation parameter (Bcast/3D/16KiB)

| Name  | Value            |
|-------|------------------|
| $np1$ | 8 [node]         |
| $np2$ | 6 [node]         |
| $np3$ | 8 [node]         |
| $B$   | 13,500 [MB/s]    |
| $m_B$ | 16,384 [byte]    |
| $L$   | 1.600 [ $\mu$ s] |
| $t_o$ | 8.370 [ $\mu$ s] |

評価に用いた各パラメータを表5に示す。通信時間の見積式から算出された通信性能は表1の通りである。通信時間の見積式と測定のフィッティングによるスループットの比較は図5の通りである。点線が見積式による結果である。

ピーク性能  $B$  は、通信バンド幅の3方向分の実効値 13,500[MB/s]に比べ、測定結果では 10,400[MB/s]と約 23% 低く1次元及び2次元と同様の傾向である。遅延時間  $t_L$  は、通信時間の見積式から求めた値が 115[ $\mu$ s]に比べ測定結果からの値は 169[ $\mu$ s]と約 47%増加している。3次元では1次元及び2次元と比べると、パイプラインのステップ数が 383 (1次元), 69 (2次元) から 20 (3次元) へと更に減少するため、遅延時間が 2,795[ $\mu$ s] (1次元), 548[ $\mu$ s] (2次元) から 169[ $\mu$ s] (3次元) へ減少した。

#### 4.2.4 Allreduce (3次元)

##### (1) 測定結果のフィッティング

Allreduce について Trinaryx3 の効果がある3次元ノード割当てについて評価を行った。8×6×8=384 ノード (3次元ノード割当て) において、Bcast と同様にパイプライン転送のセグメントサイズを 16[KiB]と固定として Allreduce の測定を行い、128[KiB]以降のデータを用いて、最小二乗法によるフィッティングを行った。Allreduce は内部で Reduce 処理と Bcast 処理を行うため、スループットの算出には、実際に通信されるメッセージサイズ  $M$  が2倍であることを用いた。表1に求められた通信性能を、図6にスループットの測定結果とフィッティング結果を示す。丸印が測定結果で実線がフィッティング結果である。

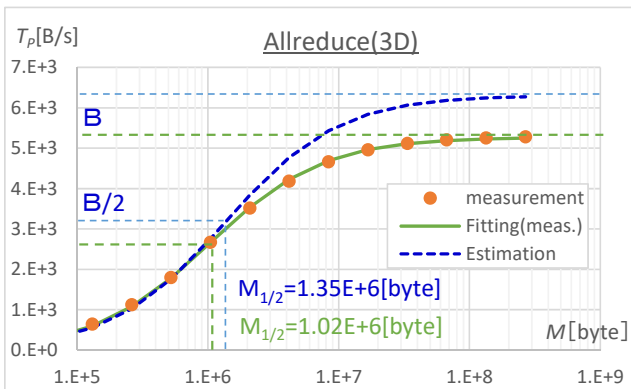


図6 通信時間の測定結果と見積式から算出したスループットの比較 (Allreduce/3D/16KiB)

Figure 6 Comparison of throughput calculated from measurement and estimation (Allreduce/3D/16KiB)

##### (2) 見積式からの通信時間の算出

通信時間  $t$  の内訳は、Bcast 処理の時間  $t_B$ 、Reduce 処理の時間  $t_R$  であるとする。 $t_B$  は、式(14)の Bcast (3次元) の見積式で記述できる。 $t_R$  は、Bcast 処理の  $t_B$  の見積式に、Reduce 処理の演算スループット  $T_C$  (=演算データ容量/処理時間)

を考慮することで算出可能である。通信時間の式を以下に示す。MPI レイテンシ  $L$  は各通信段階で必要であり、1要素の allreduce などによるオーバーヘッド  $t_O$  は、Bcast 処理の前に1回だけ必要である。尚、セグメントサイズを、Bcast 処理で  $m_B$ 、Reduce 処理用で  $m_R$  とする。

$$t = t_B + t_R \quad (17)$$

$$t_B = ((np1 - 1) + (np2 - 1) + (np3 - 1) + 1) \times (L + m_B / (B/3)) + ((M/3) / m_B) \times \max(L, m_B / (B/3)) + L + t_O \quad (18)$$

$$t_R = ((np1 - 1) + (np2 - 1) + (np3 - 1) + 1) \times (L + m_R / (B/3) + m_R / (T_C/3)) + ((M/3) / m_R) \times \max(L + m_R / (T_C/3), m_R / (B/3)) + L \quad (19)$$

これらの式からメッセージサイズ  $M$  に依存しない項を抽出すると測定に用いたセグメントサイズの条件で、 $L < m_B / (B/3) < L + m_R / (T_C/3)$  となり、 $t_L$  は以下となる。

$$t_L = (np1 + np2 + np3 - 2)(L + m_B / (B/3)) + (np1 + np2 + np3 - 2)(L + m_R / (B/3) + m_R / (T_C/3)) + 2L + t_O \quad (20)$$

一方、Reduce と Bcast 処理によりデータ転送量は2倍となるため、実効のピーク性能  $B'$  は以下のように書き下せる。

$$t = t_L + 2M / B' \quad (21)$$

$$B' = 2 / (1/B + 1/T_C + 1/(3m_R/L)) \quad (22)$$

よってスループット  $T_p$  は以下となる。

$$T_p = 2M / t = 2M / (t_L + 2M / B') = B' / (1 + (B' t_L) / 2M) \quad (23)$$

スループットは式(3)で記述されるため、 $M_{1/2}$  は  $B'$  を用いて、

$$M_{1/2} = B' t_L / 2. \quad (24)$$

ここで、演算スループット  $T_C$  は「京」においては、メモリバンド幅  $B_M$  (実効値は 46,000[MB/s]) から通信処理で利用する分を除いて算出され以下の式と値となる。

$$T_C = (B_M - 2B) / 4 = 4,750[MB/s] \quad (25)$$

ここで係数2は受信と送信による効果であり、係数4は変

表6 見積式に用いた評価パラメータ (Allreduce/3D/16KiB)

Table 6 Evaluation parameter (Allreduce/3D/16KiB)

| Name  | Value            |
|-------|------------------|
| $np1$ | 8 [node]         |
| $np2$ | 6 [node]         |
| $np3$ | 8 [node]         |
| $B$   | 13,500 [MB/s]    |
| $m_B$ | 16,384 [byte]    |
| $m_R$ | 16,384 [byte]    |
| $L$   | 1.600 [ $\mu$ s] |
| $T_C$ | 4,750 [MB/s]     |
| $t_O$ | 8.370 [ $\mu$ s] |

数バッファの Load/Store 数である。評価に用いた各パラメータを表 6 に示す。通信時間の見積式から算出した通信性能は表 1 の通りである、通信時間の見積式と測定結果のフィッティングによるスループットの比較は図 6 の通りである。点線が見積式による結果である。

ピーク性能  $B$  は、通信バンド幅の 3 方向分の実効値 13,500[MB/s]に対して、Reduce 処理と Bcast 処理の両方を実行するため、通信時間の見積式で実効 6,306[MB/s]となる。測定結果は 5,266[MB/s]であり、見積式に比べ約 16%の違いがある。遅延時間  $t_L$  は、見積式から求めた値が 428[μs]に比べ、測定結果の値は 389[μs]と約 9%の違いがある。

#### 4.2.5 Allgather (3 次元)

##### (1) 測定結果のフィッティング

Allgather 通信について、Tofu 専用アルゴリズム 3D-multiring [5]が適用される 3 次元ノード割当について評価した。8×6×8=384 ノード(3 次元ノード割当)において、Allgather の測定を行い、最小二乗法によるフィッティングを行った。Allgather は各ランクが持つデータを全ランクへ転送するため、スループットの算出には、実際のメッセージサイズがノード数倍であることを用いた。また通信サイズにより通信時間の見積式が変わることが想定されるため、32[KiB]までと 64[KiB]以降の 2 つの領域についてそれぞれフィッティングを行った。表 1 に求められた通信性能を、図 7 にスループットの測定結果とフィッティング結果のグラフに示す。丸印が測定結果で実線がフィッティング結果である。破線は、フィッティング領域の境界である。

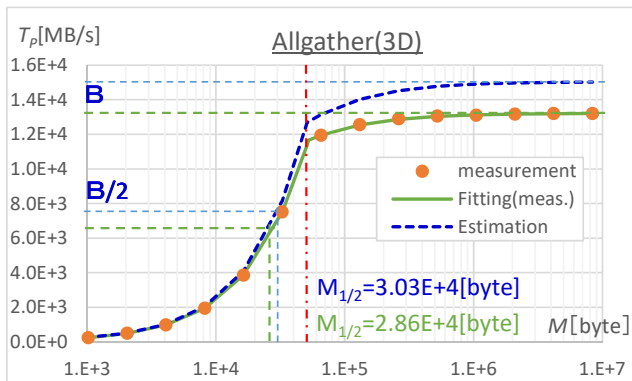


図 7 通信時間の測定結果と見積式から算出したスループットの比較 (Allgather/3D)

Figure 7 Comparison of throughput calculated from measurement and estimation (Allgather/3D)

##### (2) 見積式からの通信時間の算出

3D-multiring は、データを 4 分割して 4TNI を使った 4 方向の同時通信を行い、3 次元方向を 3 段階に分けてリング状に転送する方式である[5]。まず 1 段階目で 3 次元の各軸方向に転送して、2 段階目で 1 方向分のデータを次の軸方

向に転送し、3 段階目で 2 方向分のデータを次の軸方向に転送する。1 方向当りの通信では、メッセージサイズが  $M/4$ 、通信バンド幅が  $B/4$  となり、データ転送の時間は  $(M/4)/(B/4)=M/B$  となる。MPI レイテンシにはソフトウェアとハードウェアそれぞれの通信の準備が含まれる。4TNI を使った同時通信ではソフトウェアの通信の準備を 4 回行うため、ここでは MPI レイテンシをソフトウェア部分  $L_S$  とハードウェア部分  $L_H$  に分ける。各段階における転送方向のノード数を  $np1, np2, np3$  (総ノード数  $np = np1 \times np2 \times np3$ ) とおくと、通信時間の見積式は以下となる。

$$\begin{aligned} t = & (4L_S + L_H + M/B) \\ & + \max(4L_S, L_H + M/B) (np1 - 1 - 1) \\ & + (4L_S + L_H + M/B) \\ & + \max(4L_S, L_H + M/B) (np1 \times np2 - np1 - 1) \\ & + (4L_S + L_H + M/B) \\ & + \max(4L_S, L_H + M/B) (np - np1 \times np2 - 1) \\ = & 3(4L_S + L_H + M/B) \\ & + \max(4L_S, L_H + M/B) (np - 4) \end{aligned} \quad (26)$$

この式をメッセージサイズ  $M$  依存する項と依存しない項に分けると実効のピーク性能  $B'$  並びに遅延時間  $t_L$  は以下となる。

1)  $4L_S > L_H + M/B$  ( $M < B(4L_S - L_H)$ ) : 短メッセージ時

$$t_L = 3(4L_S + L_H) + 4L_S(np - 4) \quad (27)$$

$$B' = B np/3 \quad (28)$$

2)  $4L_S \leq L_H + M/B$  ( $M \geq B(4L_S - L_H)$ ) : 長メッセージ時

$$t_L = 3(4L_S + L_H) + L_H(np - 4) \quad (29)$$

$$B' = B np/(np - 1) \quad (30)$$

但し、実効のピーク性能  $B'$  の導出には、実際に送受信されるメッセージサイズが  $np$  倍であることを用いて算出した。ピーク性能となる最大スループット  $B_{max}$  は長メッセージ領域の見積式で記述され、以下となる。

$$B_{max} = B np/(np - 1) \quad (31)$$

「京」における 4TNI 分の総通信バンド幅は最大 15,000[MB/s]である[5]。MPI レイテンシのソフトウェア部分  $L_S$  は 4[byte]の通信時間が近似的に  $4L_S(np-1)$  となることから算出し、MPI レイテンシ 1.6[μs]から引いた値をハードウェア部分  $L_H$  とした。評価に用いた各パラメータを表 7

表 7 見積式に用いた評価パラメータ (Allgather/3D)

Table 7 Evaluation parameter (Allgather/3D/16KiB)

| Name  | Value         |
|-------|---------------|
| $np1$ | 8 [node]      |
| $np2$ | 6 [node]      |
| $np3$ | 8 [node]      |
| $B$   | 15,000 [MB/s] |
| $L_S$ | 1.0 [μs]      |
| $L_H$ | 0.6 [μs]      |

に示す。また各メッセージサイズ領域にて求めた通信性能は表 1 の通りである。通信時間の見積式と測定結果のフィッティングによるスループットの比較は図 7 の通りである。点線が見積式による結果である。

スループット  $T_p$  はメッセージサイズ  $M$  により 2 種類の評価式に従う。評価パラメータからこのメッセージサイズを見積もると約 49.8[KiB] となり、32[KiB] と 64[KiB] の間で評価式が切り替わることが分かる。

ピーク性能  $B$  は、短メッセージ領域での実測値と見積式での値とに大きな差があるが、 $B$  は長メッセージの極限で達成される外挿値であるため、短メッセージ領域の評価に大きな問題はない。長メッセージ領域での通信時間の見積式で算出された値は、15,040[MB/s] であるのに対して、測定結果では 13,210[MB/s] であり、見積式に比べ約 12% の違いである。遅延時間は、見積式から求めた値は、各メッセージ長領域で 1,534[μs] と 242[μs] であるのに対し、測定結果での値は 1,590[μs] と 201[μs] で約 4%~17% の違いである。

#### 4.3 通信性能のまとめ

今回は、「京」での MPI 通信性能のスループットの実測値について、通信モデルから通信時間の見積式を理論的に導出し、ピーク性能と遅延時間を用いた評価を行った。

PingPong と 1 次元の Bcast では、ピーク性能は Bcast が 20% 程低く、他の集団通信 Allreduce と Allgather でも同様の傾向であった。遅延時間としてパイプライン転送の立ち上げ時間が余計にかかるため Bcast での遅延時間は大きくなる。図 8 に PingPong と Bcast の測定結果から得られたスループットの比較を示す。実線が PingPong で点線が Bcast である。

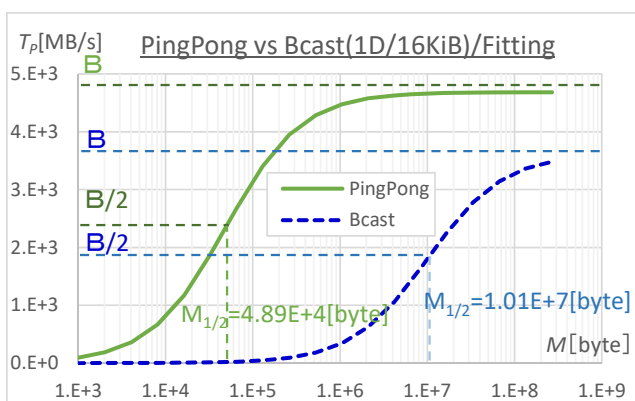


図 8 PingPong と Bcast(1D/16KiB)のスループット

Figure 8 Throughput of Bcast (1D,2D,3D/16KiB)

図 9 は、Bcast について 1 次元、2 次元、3 次元での測定結果のフィッティングしたスループットを比較したものである。実線が 1 次元、点線が 2 次元、破線が 3 次元である。

ジョブ形状の次元が上がると多方向の同時通信が可能となり、ピーク性能  $B$  が 1 次元に比べて 2 次元で約 2 倍、3 次元で約 3 倍となる。また、次元が上がると、パイプラインのステップ数が小さくなるため立ち上がりに必要な時間が小さくなり、遅延時間  $t_L$  が小さくなる。つまりジョブのノード割り当てを、より高次元で使用した方が、通信性能が高いことが分かる。しかし実際には、短いメッセージでの通信において、通信時間の最適化のためセグメントの長さが変化し、かつ分割のレイテンシも増大するため、より汎用的な Open MPI アルゴリズムが選択されることが想定される。このため通信性能に差異がなくなるか、逆転が起り得る可能性もある。

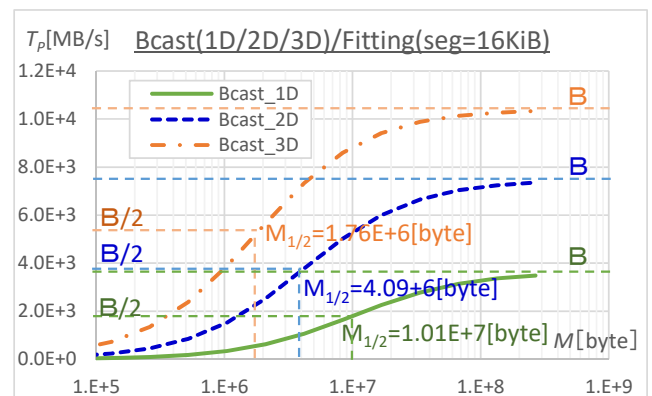


図 9 Bcast(1D,2D,3D/16KiB)のスループット

Figure 9 Throughput of PingPong and Bcast (1D/16KiB)

#### 5. 展開と今後の課題

今回評価した MPI 集団通信の実測値と通信時間の見積式によるスループットの傾向について、ピーク性能で 10%~20% 程度、遅延時間で 10%~50% の精度で説明することができた。特に Tofu インターコネクトに最適化されたアルゴリズムが、想定される性能をある程度達成できていることを確認することができたと言える。これら性能の差異の原因として、ハードウェアに起因するパラメータの見積もりが不十分である可能性や、通信ライブラリの実装方法による性能の変化などが可能性として考えられる。特に、Bcast などの集団通信全般では、ピーク性能が理論バンド幅と比べて 10% 程度低く実測されているが、双方向通信による影響と推察している。今後、更なる検証を行うことで、通信性能向上に向けた課題を見だし、通信アルゴリズムやライブラリ開発などへのフィードバックへつなげていくことが期待される。

また、今回報告した以外の集団通信の性能についても Bcast, Allreduce, Allgather の見積式を拡張して評価することで、より多くの知見を得ることが出来ると考えられる。

**謝辞** 本報告に際し、理化学研究所計算科学研究機構運用技術部門、富士通株式会社 SE の諸氏に感謝します。本論文の結果は、理化学研究所計算科学研究機構が保有するスーパーコンピュータ「京」によるものです。

## 参考文献

- [1] Intel® MPI Benchmarks User Guide and Methodology Description, <https://software.intel.com/en-us/articles/intel-mpi-benchmarks>
- [2] 北澤好人, 黒田明義, 南一生, 庄司文由: スーパーコンピュータ「京」における MPI 通信性能の評価, 研究報告ハイパフォーマンスコンピューティング(HPC), 2015-HPC-153(34), pp.1-7 (2016-2-23).
- [3] Takashi Toyoshima, ICC: An interconnect controller for the Tofu interconnect architecture., Hot Chips 22 (2010).
- [4] 安島雄一郎, 井上智宏, 平本新哉, 清水俊幸: スーパーコンピュータ「京」のインターコネクト Tofu, FUJITSU.63,3(05,2012), pp.260-264, <http://www.fujitsu.com/downloads/JP/archive/imgjp/jmag/vol63-3/paper05.pdf>
- [5] Tomoya Adachi, Naoyuki Shida, Kenichi Miura, Shinji Sumimoto, Atsuya Uno, Motoyoshi Kurokawa, Fumiyoshi Shoji and Mitsuo Yokokawa: The Design of Ultra Scalable MPI Collective Communication on the K Computer., 2013 Comput. Sci., 28(2-3), pp.147-155 (May 2013).
- [6] Yuichiro Ajima, Yuzo Takagi, Tomohiro Inoue, Shinya Hiramoto and Toshiyuki Shimizu: The Tofu interconnect., Proc. of HotI 2011, pp.87-94 (Aug. 2011).
- [7] 松本幸, 安達知也, 住元真司, 南里豪志, 曾我武史, 宇野篤也, 黒川原佳, 庄司文由, 横川三津夫: MPI\_Allreduce の「京」上での実装と評価, 情報処理学会誌, コンピューティングシステム, Vol.5, No.5, pp.152-162 (Oct.2012).
- [8] Roger W. Hockney: The Science of Computer Benchmarking, ISBN-13: 978-0898713633, section 3.3, pp42-47 (1996).