Vol.2020-HPC-177 No.23
2020/12/22

MPIプログラムにおける遅延挿入による不規則な多対多通信の効率化

山田 広俊[1,a)]　置田 真生[1,b)]　伊野 文彦[1,c)]

概要：本報告の目的は，MPI プログラムにおける 1 対 1 通信を用いた不規則な多対多通信の高速化である．プロセスごとに異なる数の送受信命令を実行する不規則な多対多通信では，規則的な場合と比較して，MPI の集合通信を用いると実行効率が低下する．その理由は，通信の競合が不規則に発生し，通信のスループットが低下するためである．本報告では，スイッチ接続および Eager プロトコルを前提に，競合を回避する通信スケジュールの決定手法を提案する．提案手法のアイデアは，一部の送信命令の前に遅延を挿入し，競合を回避するようなトラフィックを誘導することである．実験の結果，既存手法を不規則な多対多通信に単純に適用した場合と比較して，提案手法は多対 1 通信に近い通信パターンに対して実行時間を最大 6.3 倍高速化した．また，MPI の集合通信命令と比較して，提案手法は全対全の通信パターンに対して 3.0 倍の高速化を得た．

1. はじめに

多対多通信の高速化が，Message Passing Interface [1]（MPI）を用いたプログラムにおける並列化効率向上のために重要である．多対多通信は 2 通りに分類できる．1 つ目は，各 PE が同数の送受信命令を実行し，かつ全 PE で同数の送信命令を実行する，規則的な多対多通信である．規則的な多対多通信は，密行列積やステンシル計算 [2] に用いられる．2 つ目は不規則な多対多通信である．不規則な多対多通信は，疎行列積 [3] やグラフ処理 [4] のために必要である．

多対多通信の最適化のためには，通信路の競合を回避する条件のもとで，実行時間を最小化する必要がある．規則的な多対多通信に対しては，最適なアルゴリズムを実装した MPI の集合通信が存在する [5]．一方，不規則な多対多通信に対しては，最適化の研究が不十分である．その理由は，通信の競合が不規則に発生し，かつネットワークトポロジや通信プロトコルに依存して競合の発生条件が異なるためである．

不規則な多対多通信に対しては，1 対 1 通信を組み合わせて，最適な通信スケジュールを決定する必要がある．通信スケジュールの最適化は，各 PE における送信命令の開始順序を決定する組合せ最適化問題に帰着する．

これまでに，競合を回避するスケジューリング手法が存在する．Thakur ら [5] は，各 PE におけるネットワーク上の距離が等しい条件の下，全対全通信の競合を完全に回避する手法を提案した．この手法は，まずリング通信パターンを作成し，次にリングの送信先をシフトしながら競合を回避する通信パターンを段階的に作成する．Karwande ら [6] は，バリア同期の挿入による通信の分断によって，競合を回避する手法を提案した．この手法は，可変量データを送信する任意の多対多通信の競合を完全に回避する．ただし，この手法は多対多通信の実行時間を最小化しない，ヒューリスティックな手法である．

しかし，既存手法は 1 つの PE に受信が集中する不規則な多対多通信に対して，非効率的な通信スケジュールを作成する．Thakur らの手法を単純に適用すると，リングおよびそのシフト通信パターンが作成できず，送信が 1 つの PE に集中するために競合が発生する．Karwande らの手法は，競合を避けるために送信が逐次化し，かつ送信のたびにバリア同期を必要とするため，バリア同期のオーバヘッドが増大する．

そこで本報告は，任意の不規則な多対多通信に対して，効率的な通信スケジュールを決定する手法を提案する．提案手法が決定するスケジュールは，バリア同期を挿入せずに全ての競合を回避する．提案手法のアイデアは，一部の送信命令の前に遅延を挿入することで，競合を回避するよ

[1] 大阪大学 大学院情報科学研究科
Graduate School of Information Science and Technology, Osaka University
[a)] h-yamada@ist.osaka-u.ac.jp
[b)] okita@ist.osaka-u.ac.jp
[c)] ino@ist.osaka-u.ac.jp

ⓒ 2020 Information Processing Society of Japan

うなトラフィックを誘導することである．提案手法はまず，LogGP モデル [7] に基づいて，多対多通信の競合の発生および実行時間を予測するモデルを作成する．予測モデルはスイッチ接続および Eager プロトコルを前提とする．次に，予測モデル上の全ての競合を回避した上で，実行時間を削減する通信スケジュールを決定する．実行時間削減のために，遅延の挿入回数を削減する．提案手法は，遅延の挿入回数の最小化を保証しないヒューリスティック手法である．

提案手法は，MPI 実装が最適化する特定の通信パターンに対してだけでなく，任意の多対多通信のパターンに対して競合を回避するスケジュールを作成できる．特に，不規則な多対多通信に対しては `MPI_Alltoallv` による実装と比較して，スイッチ接続トポロジにおいて静的な通信パターンを繰り返す場合に有用性が高い．

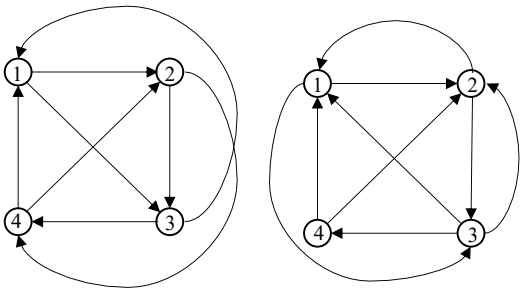
以降，2 節では関連研究を提案手法と比較する．3 節では通信スケジュールに関する用語を定義し，4 節では本報告が解決を目指す問題を定義する．5 節では提案手法を説明し，6 節では提案手法を評価する．最後に，7 節で本報告をまとめる．

2. 関連研究

Thakur ら [5] は，全対全通信を対象に，リング通信パターンを利用した競合の回避手法を提案した．この手法は，リングおよびリングの送信先をシフトした通信パターンを段階的に実行する．各段階の通信パターンが競合を回避するために，結果的に全ての競合を回避する．この手法は，全 PE 間のネットワーク上の距離が等しい前提のもと，規則的な多対多通信を最適化する．不規則な多対多通信は，全対全通信における一部の送信命令を除外したものである．Thakur らの手法を不規則な多対多通信に対して単純に適用すると，除外した部分に次の送信命令が繰り上がることにより，競合が発生する．提案手法は不規則な多対多通信においても，競合を完全に回避する．

Karwande ら [6] は，可変量データにおける不規則な多対多通信を対象に，競合の完全な回避手法を提案した．この手法は，スイッチ接続を前提としたアルゴリズムである．Karwande らアイデアは，バリア同期の挿入によって，競合を引き起こす通信を分断することである．この手法は，全対全通信に近いパターンに対して有効である．この理由は，全対全通信に近いパターンは，バリア同期による各 PE の待機時間を削減できるためである．この手法を多対 1 通信に近い通信パターンに対して適用すると，送信が逐次化するために，バリア同期のオーバヘッドが増加する．また，提案手法はバリア同期を挿入せずに競合を完全回避する．

Jelena ら [8] は，MPI の集合通信で使用されているアルゴリズムを，LogGP モデル [7] に基づいて比較した．ただしこの研究が比較するアルゴリズムは，不規則な多対多通信は対象外である．提案手法は，任意の多対多通信を対象とした実行時間予測モデルを，LogGP モデルに基づいて作成する．



(a) 規則的　　(b) 不規則

図 1: 多対多通信の例

3. 諸定義

本稿における多対多通信の定義を，全 PE が自身を除く各 PE に対して高々 1 回データを送信するような集合通信と定める．例えば，MPI の主な集合通信，`MPI_Alltoall`，`MPI_Gather` および `MPI_Scatter` は，多対多通信である．さらに，**図 1** に示すような MPI 集合通信のパターンに対応しない多対多通信も本稿の対象である．本稿では，通信データの内容や通信に伴う計算は考慮せず，通信パターンのみに着目する．

3.1 通信パターン

多対多通信の通信パターンを，PE を頂点とする単純有向グラフ $G = (P, A)$ で表す．ここで，$P = \{1 \dots n\}$ は頂点集合を表し，有向辺 $(p, q) \in A$ は PE p から PE q への通信を表す．G は単純グラフである．すなわち，送信元 PE と送信先 PE は必ず異なり，かつ送信先 PE と受信先 PE の組は固有である．なお簡単のため，G は連結グラフを想定する．非連結である場合は，連結成分ごとに独立してスケジュールを決定すればよい．

また，PE $i \in P$ における送信パターン A_i を，式 (1) のように，頂点 i における出次辺の集合 N_i^+ で表す．

$$A_i = N_i^+ = \{(p, q) \in A \mid p = i\} \tag{1}$$

G が単純グラフであることから，$|A_i| \leq n - 1$ である．

ここで，規則的な多対多通信を，各プロセスにおいて送信命令と受信命令の数が等しく，かつ全プロセス間で送信命令の数が等しい多対多通信と定義する．したがって，規則的な多対多通信の通信パターン G は以下の条件式 (2) を満たす．

$$\exists k \in \mathbb{N}, \forall i \in P,\ |N_i^+| = |N_i^-| = k \tag{2}$$

なお，N_i^- は頂点 i の入次辺の集合を表す．例えば，`MPI_Alltoall` および**図 1a** は規則的な多対多通信である．

一方，条件式 (2) を満たさない多対多通信を，不規則な多対多通信と定義する．例えば，`MPI_Gather` および `MPI_Scatter` は不規則な多対多通信である．**図 1b** は各頂点における入次数と出次数が異なり，不規則な多対多通信である．

3.2 通信スケジュール

まず，ある PE $i \in P$ に対する送信スケジュール B_i を，A_i の全要素に対する順序付けと定義する．すなわち，B_i は辺 $a \in A$ の系列であり，式 (3) で表す単射と等価である．

$$B_i : \{1, 2, \ldots, |A_i|\} \to A_i \tag{3}$$

次に，多対多通信の全体スケジュール B を，全プロセスの送信スケジュールの系列と定義し，式 (4) で表す．

$$B = (B_1, B_2, \ldots, B_n) \tag{4}$$

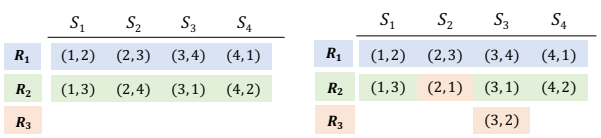
なお，各プロセスの受信スケジュールは，受信すべきデータの到着順に従って決定する同時に到着するデータが複数存在する場合，それらの受信順序は任意とする．

スケジュール B の具体例を，Thakur らの手法 [5] を用いて示す．Thakur らの手法は，まず全プロセスの最初の送信命令の組すなわち $R_1 = (B_1(1), B_2(1), \ldots, B_n(1))$ を，R_1 がリングを形成するように選択する．複数の組がリングを形成しうる場合は，任意の 1 つを選択する．次に，R_1 で形成したリングを d シフトして得られる組から，順序未定の部分集合（$A \setminus R_1$）に該当する組を選択し，R_2 とする．ただし d シフトとは，R_1 の各要素 (p, q) を $(p, ((q + d - 1) \mod n) + 1)$ に変更して得られる組を指す．同様に R_k $(2 < k \leq n)$ に対して繰り返すことで，B を決定する．**図 2a** に，図 1a の規則的な多対多集合に Thakur らの手法を適用した例を示す．R_1 はリングを形成しており，R_2 は R_1 を 1 シフトした組である．2 シフトすると R_3 であるが，該当する通信が存在しない．

不規則な多対多通信を，規則的な多対多通信の一部を削除した場合と解釈することで，Thakur らの手法を用いてスケジュールを決定できる．具体的には，不規則な多対多通信 G を内包する規則的な多対多通信 G' に対して Thakur らの手法を適用してスケジュール B' を作成し，G に含まれない通信を B' から取り除くことで B を得る．この方法を用いて，図 1b の不規則な多対多通信に Thakur らの手法を適用した例を**図 2b** に示す．R_2 の一部が存在しないため，$(2, 1) \in R_3$ が繰り上げられて $B_2(2) = (2, 1)$ となる $B_2(2) = (2, 1)$ および $B_3(2) = (3, 1)$ の送信先 PE が等しい，つまり PE 2 および PE 3 が同時に PE 1 に送信するため，スイッチ接続の場合に通信の競合が発生する．

4. 問題の定義

本稿で対象とする問題は，多対多通信 G のメイクスパン $M_G(B)$ を目的関数とする制約つき最小化問題である．この問題を，B を決定変数とする以下の最小化問題と定義する．



(a) 規則的　　(b) 不規則

図 2: Thakur らの手法を用いたスケジュールの例

$$\text{最小化} \qquad M_G(B) \tag{5}$$

$$\text{制約} \quad a_1 \neq a_2 \Rightarrow \neg C(a_1, a_2), \forall (a_1, a_2) \in A \times A. \tag{6}$$

ここで，$C(a_1, a_2)$ は，通信 a_1 および a_2 の対が通信の競合を引き起こす場合に真となる述語とする．したがって，制約条件 (6) は，B において競合が発生しないための条件である．すなわち，競合を完全に回避したうえで，メイクスパンが最小となるスケジュール B を求める．

メイクスパン $M_G(B)$ は，受信処理の完了が最も遅い通信の完了時刻に等しい．したがって，通信 $a \in A$ のスケジュール B における受信完了時刻 $T_B(a)$ を用いて，式 (7) でメイクスパンを定義する．

$$M_G(B) = \max\{T_B(a) \mid a \in A\} \tag{7}$$

このメイクスパン最小化問題は，特定の通信パターンに対して実行可能解を持たない．すなわち，競合の発生を回避できない通信パターンが存在する．例えば，`MPI_Gather` の単純な実装のように，ある PE p に他の全 PE が 1 回ずつ送信を行う通信パターンに対しては，通信スケジュールは一意に決定しかつ必ず競合を含む．

5. 提案手法

任意の不規則な多対多通信に対して，通信の競合を完全に回避したうえでメイクスパンを短縮するスケジューリング手法を提案する．まず，各 PE の通信スケジュールに遅延の挿入を許すことで，任意の通信パターンに対する競合の回避を可能とする．次に，LogGP モデル [7] に基づいて，競合の発生と各通信命令の終了時刻を予測するモデルを導入する．最後に，遅延挿入の回数をヒューリスティックに削減するスケジューリングのアルゴリズムを提案する．

なお，通信の競合の発生条件は，ネットワークの諸条件に依存して異なる．条件を明確にするために，本稿では以下の前提が成り立つ場合に対象を限定する．

- ネットワークのトポロジは単一のスイッチ接続である．
- 双方向通信が可能である．
- 全ての通信において，データの転送時間は等しい．
- MPI の通信プロトコルは Eager を用いる．

ⓒ 2020 Information Processing Society of Japan

5.1 遅延挿入による競合回避

競合する送信命令の開始を遅延することで，通信の競合を回避する．通信の競合が発生する原因は，ある通信路を 2 つ以上のトラフィックが同時に使用するためである．したがって，通信路上のトラフィックが高々 1 つになるよう意図的に送信命令の開始を遅延することで，任意の通信パターンに対して競合を完全に回避できる．ただし，遅延によって全体の処理時間が増大する可能性がある．例えば，全ての通信を逐次化するように遅延することで原理的に競合を完全に回避できるが，$\mathcal{O}(|\mathcal{A}|)$ の処理時間が必要となる．したがって，競合を完全に回避するための最小の遅延を求める必要がある．

遅延を表現できるように，通信スケジュールの定義を拡張する．PE i の拡張送信スケジュール S_i を，式 (8) で定義する．

$$S_i : \{1, \ldots, n\} \to A_i \cup \{\delta\} \tag{8}$$

ここで，δ は $|\delta|$ 時間の遅延を表す．実装には `nanosleep(2)` を用いる．例えば，図 2b に示す S_2 の 2 番目に遅延を挿入したスケジュール S_2 は，$S_2 = ((2,3), \delta, (2,1))$ となる．単一スイッチ接続かつデータ転送時間が均一な場合，全対全通信に Thakur らの手法 [5] を用いると競合のないスケジュールとなるため，$|S_i| \leq n$ を満たす遅延挿入で競合の完全回避が可能である．

したがって，提案手法が解くメイクスパン最小化問題を式 (9) に拡張し，目的関数を変更する．

$$\begin{aligned} &\text{最小化} && M_G(S) && (9)\\ &\text{制約} && a_1 \neq a_2 \Rightarrow \neg C(a_1, a_2), \forall (a_1, a_2) \in A \times A. \end{aligned}$$

ここで S は，各 PE の送信スケジュールの系列 $S = (S_1, S_2, \ldots, S_n)$ である．制約条件に変更はない．

5.2 多対多通信の実行時間予測モデル

通信スケジュールを入力とし，競合の発生と各通信命令の終了時刻を予測するモデルを導入する．予測モデルにおける 1 対 1 通信の挙動は LogGP モデルに基づき，Eager プロトコルおよびスイッチ接続を前提とする．なお，各送受信命令の発行時間は 0 とみなす．予測モデルにおける変数は，以下の 5 つである．

- n : PE 数.
- k : データ量.
- $I(k)$: 送信間隔．送信命令を開始してから，次の命令を開始できるようになるまでの時間を表す．LogGP モデルの変数を用いて，$I(k) = o + (k-1)G$ と表す．
- L : レイテンシ．データが送信元 PE を出発してから，送信先 PE に到着するまでの時間を表す．LogGP モデルのレイテンシ L と同一の値である．
- o : 受信オーバーヘッド．到着したデータを受信処理する時間を表す．LogGP モデルのオーバヘッド o と同一の値である．

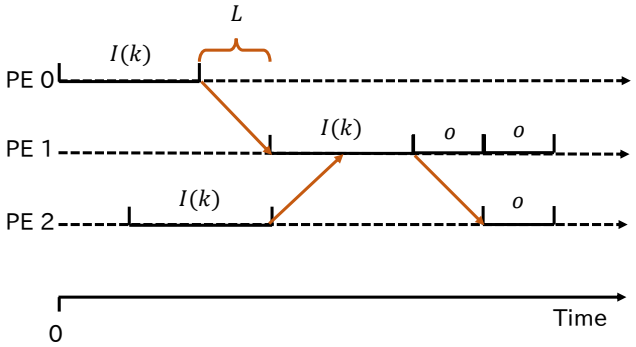


図 3: 予測モデルにおける送信命令の実行時間

図 3 に，予測モデルにおける送信命令の処理をガント図として表示する．矢印は，データの送信完了から到着までの処理を示す．縦軸は PE を，横軸は時間を表す．

1 対 1 通信 (p, q) の処理時間は，データ到着時点における送信先 PE の状態によって増減する．PE q にデータが到着した時点で，PE q の送信処理または (p, q) より先に到着したデータの受信処理が未完了の状態をビジー状態と定義する（図 3 の $(0, 1)$ および $(2, 1)$）．PE q がビジー状態ではない場合を，非ビジー状態と定義する（図 3 の $(1, 2)$）．PE q が非ビジー状態の場合，PE q はデータの到着直後に受信処理を開始する．したがって，送信先 PE が非ビジー状態の場合における 1 対 1 通信の処理時間は，式 (10) となる．

$$I(k) + L + o \tag{10}$$

一方，PE q がビジー状態の場合，PE q の送信処理または (p, q) より先に到着したデータの受信処理が完了するまで，PE q は (p, q) の受信処理を開始しない．PE q にデータが到着してから受信処理を開始するまでの時間を w とすると，PE q がビジー状態の場合における 1 対 1 通信の処理時間は式 (11) となる．

$$I(k) + L + w + o \tag{11}$$

予測モデルにおける競合の発生は，2 つの送信命令の開始時刻に依存して決まる．2 の送信命令 $a_1 = (p_1, p_1)$ および $a_2 = (p_2, p_2)$ の開始時刻は，$(S_{p_1}^{-1}(a_1) - 1) \times I(k)$ および $(S_{p_2}^{-1}(a_2) - 1) \times I(k)$ である．2 つの送信命令における開始時刻の差が $I(k)$ 未満の場合に，競合が発生すると定義する．したがって，式 (12) を満たす場合に競合が発生する．

$$C(a_1, a_2) = (q_1 = q_2 \wedge |S_{p_1}^{-1}(a_1) - S_{p_2}^{-1}(a_2)| < 1) \tag{12}$$

例えば，図 3 の 2 つの送信命令 $(0, 1)$ および $(2, 1)$ が競合を起こす．

競合が発生しない条件の下での通信 $a = (p, q) \in A$ の受

信完了時刻 $T_S(a)$ は，a の送信完了時間およびレイテンシ L の大きさに依存して決まる．まず，a の送信完了時刻は式 (13) で計算できる．

$$S_p^{-1}(a) \times I(k) \tag{13}$$

データが PE q に到着するまでに，時間 L を要する．データが到着した時点における PE q の状態は，レイテンシの値に依存する．その条件は，式 (14) となる．

$$L \geq |S_q| \times I(k) \tag{14}$$

条件式 (14) を満たす場合，PE q は常に非ビジー状態である．その理由は，PE q に対して常に o 以上の時間間隔でデータが到着するからである．a は競合しないので，q へのデータの送信間隔は常に $I(k)$ 以上である．したがって，データの到着間隔も常に $o < I(k)$ 以上となる．a の受信完了時刻 $T_S(a)$ は，a の送信完了時刻（式 (13)）および非ビジー状態の送信命令の処理時間（式 (10)）より，式 (15) となる．

$$T_S(a) = S_p^{-1}(a) \times I(k) + L + o \tag{15}$$

条件式 (14) を満たさない場合，PE q は常にビジー状態である．まず，a は PE q の受信処理の開始時刻 $|S_q| \times I(k)$ まで待機する．次に，既に PE q 到着しているデータの内，a よりも到着が早いデータの受信処理を待機する．PE q に対して a よりも早く到着するデータの数を h とすると，a の受信完了時刻 $T_S(a)$ は式 (16) となる．

$$T_S(a) = |S_q| \times I(k) + (h+1) \times o \tag{16}$$

5.3 スケジューリング手法

各プロセスの送信スケジュールに遅延を挿入することで，競合を回避したスケジュール S を求める．簡単のため，送信命令 $I(k)$ を単位時間として時間を離散化し，全 PE が同期して送信を開始すると仮定する．式 (12) より，同一の送信先 PE q へのデータ転送間隔が $I(k)$ 以上であれば，q における競合を回避できる．したがって，任意の時刻 t において各 PE を送信先とする送信命令が高々 1 つであれば，競合は発生しない．この仮定に基づいて，遅延を含むスケジュール S を決定する．全 PE の送信を同期するため，遅延時間 $|\delta|$ は，1 単位時間すなわち $|\delta| = I(k)$ とする．

5.3.1 遅延挿入による競合の回避

手法の概要をアルゴリズム 1 に示す．入力は通信パターン G，出力は通信スケジュール S である．通信スケジュール決定の際，貪欲法を用いて遅延の挿入回数を削減する．具体的には時刻 $t = 0$ から時刻 $t = (n-1)$ まで 1 単位時間ごとに全 PE のスケジュールを決定し，各時刻において独立に遅延の挿入回数を削減する．時刻 t の全 PE のスケジュールを $L_t = \{S_1(t), S_2(t), \ldots, S_n(t)\}$ で表す．

時刻 t において，ある優先順（`highest_priority_PE`）に従って，各 PE のスケジュール $S_p(t)$ を決定する．PE p に対して，A_p の全要素のスケジュールが決定していれば，処理を省く（9 行目）．スケジュール未定の送信命令（$A_p \setminus S_p$）が存在する場合，以下の基準に従って $S_p(t)$ を決定する．

(1) その時点の L_t と競合を起こさない $(p,q) \in (A_p \setminus S_p)$ があれば，(p,q) を割り当てる（10 行目）．
(2) そうでない場合，つまり L_t との競合を回避できない場合，遅延を挿入する．

したがって，得られる S は競合を含まない．

PE の優先順 `highest_priority_PE` によって遅延挿入の回数が決まる．なお規則的な通信に対しては，PE の優先順に関わらず，常に遅延挿入なしで競合を回避するよう送信命令を割り当て可能である．したがって，Thakur らの手法と同等のスケジュールが得られる．

アルゴリズム 1 提案スケジューリング手法

Input: 通信パターン $G = (P, A)$
Output: 通信スケジュール $S = (S_1, S_2, \ldots, S_n)$
1: initialize $S_i = ()$ for $\forall i \in P$
2: **for** $t = 0$ **to** $n-1$ **do**
3: $L_t \leftarrow \emptyset$
4: $X \leftarrow P$ ▷ 時刻 t においてスケジュール未定の PE 集合
5: **while** $X \neq \emptyset$ **do**
6: $p \leftarrow$highest_priority_PE(G, S, L_t)
7: $X \leftarrow X \setminus \{p\}$
8: $Z \leftarrow \delta$ ▷ 番兵
9: **for** $(p,q) \in (A_p \setminus S_p)$ **do**
10: **if** $Z = \delta \wedge \bigvee_{(x,y) \in L_t} q \neq y$ **then**
11: $Z \leftarrow (p,q)$
12: $L_t \leftarrow L_t \cup \{(p,q)\}$
13: $S_p(t) \leftarrow Z$
14: **return** S

5.3.2 遅延挿入回数の回数削減

多対多通信の実行時間を最小化するためには，競合を回避した上で，遅延挿入の回数を最小化する必要がある．各時刻における通信スケジュールを独立に決定する際，その時刻における遅延挿入の回数を削減する手法を提案する．このアルゴリズムはその時刻における遅延挿入回数の最小化を保証しない，ヒューリスティック手法である．

遅延の挿入は，時刻 t において PE p に対して送信命令を割り当てる際（アルゴリズム 1 の 10 行目），PE p の送信選択肢 Q_p が 0 である場合に発生する．ここで PE p の送信選択肢 Q_p は，時刻 t の時点で開始時刻が未定の PE p における送信命令の内，競合を回避できる送信命令の数である（式 (17)）．

$$Q_p = |(A_p \setminus S_p)| - |\forall a \in (A_p \setminus S_p), \forall l \in L_t, C(a,l) = \text{true}| \tag{17}$$

時刻 t において各 PE を順番で選択する際（アルゴリズム 1 の 6 行目），PE を選択する度に各 PE の送信選択肢が減少する．そこで，選択の際に送信選択肢が 0 に近い PE を優先する．

ⓒ 2020 Information Processing Society of Japan

アルゴリズム 2 highest_priority_PE

Input: 通信パターン G，部分的に決定したスケジュール S，その時点で時刻 t に割り当てた送信命令の集合 L_t
Output: PE q

1: **for** $p = 1$ **to** $n - 1$ **do**
2: $\quad Q_p \leftarrow |A_p \setminus S_p|$
3: $\quad$ **for** $(p, q) \in (A_p \setminus L_t)$ **do**
4: $\quad\quad$ **for** $(x, y) \in L_t$ **do**
5: $\quad\quad\quad$ **if** $q = y$ **then**
6: $\quad\quad\quad\quad Q_p \leftarrow Q_p - 1$
7: **return** m s.t. $|Q_m| \in \min\{|Q_1|, \ldots, |Q_n|\}$

PE の優先順を決定する highest_priority_PE をアルゴリズム 2 に示す．入力は通信パターン G およびその時点での S および L_t である．出力は送信選択肢が最小となる PE である（7 行目）．PE p における送信選択肢の初期値は，A_p のうち未割当の送信命令の個数である（2 行目）．未割当の送信命令の内，L_t と競合する数だけ送信選択肢を減少する（4 から 9 行目）．

5.4 多対多通信の実装

多対多通信の実装には，MPI の永続化 1 対 1 通信を用いる．この理由は，同じ通信パターンを繰り返す処理を想定するためである．

1 つの PE の挙動を記述した擬似コードを**図 4** に示す．まず，MPI_Send_init を用いて，スケジュール S に従う順に送信リクエストを登録する．次に，MPI_Recv_init を用いて，受信リクエストを登録する．今回のアルゴリズムでは，受信リクエストの順序は任意である．最後に，送信リクエストにしたがって多対多通信を実行する．この際，送信リクエストの値が NULL の場合は nanosleep を用いて delta の値だけ遅延する．delta の値は δ である．

提案手法実装のために，各 PE における送信命令の開始時刻を同期する必要がある．したがって，多対多通信の実行前に MPI_Barrier を用いる．

6. 評価実験

以下の観点から提案手法を評価する．

- 予測モデル上のメイクスパン
- 実環境における有用性

比較対象として，Thakur ら [5] の手法および MPI の集合通信を用いた．

まず，Thakur ら [5] の手法を不規則な多対多通信に対して適用する場合は，3.2 節で記述した方法を用いてスケジュール B を求め，提案手法の実装（図 4）を用いて B を実行する．この方法を以降，Ring と呼ぶ．

次に，MPI の集合通信を用いる場合，不規則な多対多通信と適合する通信パターンの集合通信命令があればそれを用いる．適合する命令がない場合，MPI_Alltoallv を用いて不規則な多対多通信を実装する．MPI_Alltoallv は，可変量データの全体的通信を実現する．不規則な多対多通信は，全対全通信における一部の通信を除外したものである．したがって，除外する通信に対応する送信命令のデータ量を 0 とすることで，MPI_Alltoallv によって不規則な多対多通信を実装できる．

```
1  // 送信スケジュールの設定
2  MPI_Send_init(.., .., .., .., .., .., sreq[0]);
3  sreq[1] = NULL;    // 遅延の挿入
4  MPI_Send_init(.., .., .., .., .., .., sreq[i]);
5  // 受信スケジュールの設定
6  for (i=0; i<3; i++)
7          MPI_Recv_init(.., .., .., .., .., .., rreq[i]);
8
9  // 多対多通信の実行
10 MPI_Barrier(comm);
11 for (i=0; i<3; i++)
12         if (sreq[i])
13                 MPI_Start(sreq[i]);      // 送信命令の開始
14         else
15                 nanosleep(delta);        // 遅延
16 MPI_Startall(2,rreq);
17 MPI_Waitall(3,sreq,sstatus);
18 MPI_Waitall(2,rreq,rstatus);
```

図 4: MPI 永続通信を用いた多対多通信の擬似コード

6.1 予測モデルにおける提案手法の評価

典型的な多対多通信の通信パターンについて，評価モデル上で Ring および提案手法の性能を比較する．比較する観点は，競合発生の有無とメイクスパンである．対象とする通信パターン 4 つを以下に示す．不規則な多対多通信の通信パターンは，gather の集合または scatter の集合に分解できる．したがって，この 2 つの通信パターンに対する性能は重要である．また，規則的な多対多通信に対する提案手法の有用性を評価するために alltoall を対象とした．Ring を triangle に適用すると，競合が発生する．そこで，提案手法を triangle に適用した場合の有用性を評価する．

scatter PE 0 が自分以外の全 PE に対して，1 回データを送信する．

gahter PE 0 以外の PE が PE 0 に対して，1 回データを送信する．

alltoall 各 PE が自分以外の全 PE に対して，1 回データを送信する．

triangle 各 PE が自分のランクより小さいランクを持つ PE に対して，1 回データを送信する．

表 1 に，各通信スケジュールにおける競合発生の有無とメイクスパンを示す．ただし，競合が発生するスケジュールのメイクスパンは省略する．この理由は，予測モデルにおけるメイクスパンは競合の回避を前提とするためであ

表 1: 予測モデルにおける性能比較

通信パターン	手法	競合の有無	メイクスパン
scatter	Ring	無	$(n-1)\times I(k)+L+o$
	提案	無	$(n-1)\times I(k)+L+o$
gather	Ring	有	—
	提案	無	$(n-1)\times I(k)+L+o$
alltoall	Ring	無	$(n-1)\times I(k)+L+o$
	提案	無	$(n-1)\times I(k)+L+o$
triangle	Ring	有	—
	提案	無	$(n-1)\times I(k)+L+o$

表 2: 実験環境（64 ノード構成のクラスタ）

CPU/ノード	Intel Xeon E3-1230 3.2 GHz（4 コア）
主記憶容量/ノード	8 GB
トポロジー	単一クロスバースイッチ
インターコネクト	1G Ethernet
OS	OpenHPC 1.3 (CentOS 7.6)
コンパイラ	gcc 7.3.0
MPI	OpenMPI 3.1.0

る．比較の結果，対象とする通信パターンに対して，Ring の場合に発生する競合を提案手法は回避した．競合が発生しない通信スケジュールのメイクスパンは全て等しい．

6.2 実環境における有用性

実環境として，64 台のノードで構成する PC クラスタを用いた．**表 2** に実験環境を示す． 送信するデータ量 k を 63KiB とする．この値は，実験環境において，Eager プロトコロルを用いて送信するデータ量の最大値である．一般に，データ量が大きい程，競合による時間増加が大きい．したがって，競合回避による実行時間の削減効果を評価するために，Eager プロトコロルにおける最大のデータ量を使用した．多対多通信は 1,000 回実行して，その平均値を用いた．図 4 の 10〜18 行目の時間を計測した．

6.2.1 実環境における遅延時間の決定

実環境においては，競合が発生してもスループットが低下しない場合がある．したがって，スループットの増加が発生しないならば，提案手法における $|\delta|$ の値を $I(k)$ より削減できる．

実環境における最適な $|\delta|$ の値を求める手法を示す．最適な $|\delta|$ の値は，gather における実行時間を計測することで推定できる．任意の PE 数における通信パターンの中で，gather はスループット低下の発生確率が最も高い．したがって，gather においてスループットの低下が発生しない $|\delta|$ の値ならば，同じ PE 数の全ての通信パターンに対してスループットが低下しない．そこで，$|\delta|$ の値を変化させながら，gather における実行時間を計測する．

図 5 に，データ量 63KiB における，$|\delta|$ ごとの gather における実行時間を示す．縦軸は提案手法による gather の実行時間を示し，横軸は遅延時間を示す．$|\delta|$ を 0 から増加すると，競合が減少することで実行時間が減少する．最適な $|\delta|$ を超えると，実行時間が増加する．図 5 においては，$|\delta| = 400\,\mu s$ の場合に実行時間が最小となる．以降の実験は $|\delta| = 400\,\mu s$ を用いる．

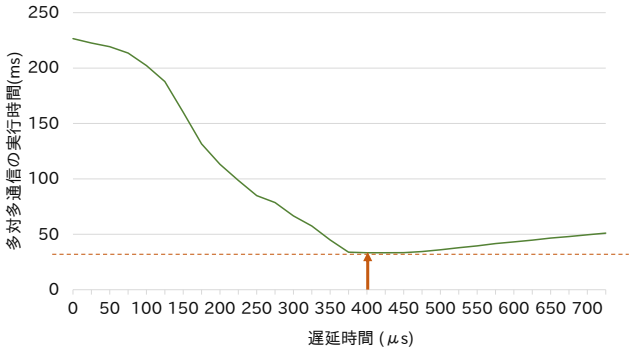


図 5: 提案手法における多対多通信時間と遅延時間の関係（gather）

6.2.2 典型的な通信パターン

典型的な多対多通信の通信パターンに対して，実環境を用いて Ring，提案手法および MPI 実装と比較する．対象とする通信パターンは，6.1 と同様である．送信スケジュールごとの実行時間を**表 3** に示す．提案手法は Ring と比較して，gather および triangle に対して 2.8 倍から 6.3 倍の高速化を得た．この理由は，遅延挿入によって実環境上の競合を回避したためと推測する．その他の通信パターンに対しては，提案手法は Ring と同等の実行時間を示した．

提案手法を MPI 実装と比較する．提案手法は alltoall に対する `MPI_Alltoall` および triangle 対する `MPI_Alltoallv` と比較して，1.1 倍から 3.0 倍の高速化を得た．一方，提案手法は scatter に対する `MPI_Bcast` と比較して，実行時間が 41.5 倍に増加した．その他の通信パターンに対しては，提案手法と MPI 実装は同等の実行時間を示した．一般に，`MPI_Alltoallv` 以外の MPI 実装はデータ量に合わせてスケジュールを最適化している．しかし，`MPI_Alltoall` 以外の通信パターンに対しても，提案手法は MPI 実装と比較して実行時間高速化する場合がある．この理由は，提案手法の実装が永続通信なためだと推測できる．MPI の集合通信は非永続通信であり，通信処理の度に送信リクエスト設定のオーバヘッドが発生する．

予測モデルの精度について考察する．典型的な通信パターンに対して，提案手法が決定する通信スケジュールのメイクスパンは全て等しい．しかし，実環境では gather および scatter に対して，triangle および alltoalll の実行時間が 1.1〜2.3 倍に増加した．提案手法は triangle および alltoall に対しては，実環境上の競合を完全には回避できないと推測する．したがって，実環境上の競合の完全回避のために，競合発生の予測精度を向上することが今後の課題

表 3: 実環境における多対多通信時間の比較（データ量 63KiB）

通信パターン	scatter				gather			alltoall				triangle		
手法	Ring	提案	MPI_Bcast	MPI_Scatter	Ring	提案	MPI_Gather	Ring	提案	MPI_Allgather	MPI_Alltoall	Ring	提案	MPI_Alltoallv
モデル上の競合	無	無	—	—	有	無	—	無	無	—	—	有	無	—
実行時間（s）	33.2	33.2	0.8	32.8	214.4	33.2	33.1	37.0	37.1	76.8	112.9	210.0	75.1	82.4

表 4: 実験に用いた scale-free の密度（頂点数 64）

パターン名	sf-1	sf-2	sf-3	sf-4	sf-5
初期頂点数 m_0[9]	4	7	12	17	28
新頂点が伴う辺数 m[9]	4	7	12	17	28
密度 D	0.12	0.20	0.31	0.40	0.50

である．

6.2.3 実用的な通信パターン

実用的な通信パターンに対して，Ring，提案手法および MPI 実装を比較する．実用的な通信パターンとして，スケールフリー性を持つグラフに対する隣接メッセージ交換を想定する．スケールフリーグラフは現実に多く存在し，隣接メッセージ交換はグラフ処理で頻繁に用いられる点で実用性が高い．以降，この通信パターンを scale-free と呼ぶ．

本稿では，密度 D が異なるスケールフリーグラフに対する実行時間を比較する．多対多通信の密度 D は，通信パターンの alltoall に対する近似度を示す指標である．密度 D は，PE 数 n の通信パターン A に対して，式 (18) によって求める．

$$D = \frac{|A|}{n \times (n-1)} \tag{18}$$

scale-free を作成するために，BA モデル [9] を基にしたアルゴリズムを用いた．ただし，本稿では BA モデルにおいて m_0 個の頂点が形成する初期グラフの辺数を 0 とした．BA モデルに基づいて作成した無向グラフにおいて，1 つの頂点に対して 1 つの PE を割り当て，隣接した頂点が双方向通信をすると仮定する．この多対多通信の密度 D は式 (19) で表す．

$$D = \frac{2 \times (n - m_0) \times l}{p \times (p-1)} \tag{19}$$

PE 数 64 における，密度 D の異なる 5 つの scale-free を作成した（**表 4**）．

scale-free に対する Ring，提案手法および MPI 実装の実行時間を計測した（**図 6**）．提案手法は Ring と比較して，sf-1，sf-2 および sf-5 に対して 1.1〜2.2 倍の高速化を得た．sf-3 および sf-4 に対して，提案手法は Ring と比較して実行時間が 1.1〜1.2 倍に増加した．また，提案手法は `MPI_Alltoallv` と比較して，全ての通信パターンに対して実行時間を削減し，1.4〜2.3 倍高速化した．

密度が低い場合に，提案手法は Ring と比較して，1.8〜2.3 倍と比較的高い高速化率を得る．この理由は，密度が低い通信パターンは gather に近似するためと推測できる．Ring は gather において競合が発生するが，提案手法は競合を回避する．したがって，密度が低い場合に提案手法の有用性が高い．

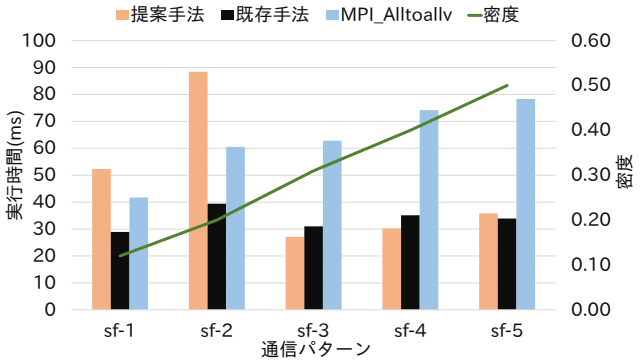


図 6: scale-free に対する多対多通信時間の比較

7. まとめ

本報告では，遅延挿入による不規則な多対多通信の効率化手法を提案した．提案手法は，(1) 通信の競合発生と多対多通信の実行時間を予測するモデルを作成し，(2) 予測モデルにおける全ての競合回避を目的とした送信スケジュールの決定手法を提案した．(1) における実行時間の予測モデルは LogGP モデル [7] に基づき，スイッチ接続および Eager プロトコルを前提とする．(2) において，特定の送信命令の開始を意図的に遅延することで，通信の競合を全て回避する．

予測モデル上で，既存手法を典型的な多対多通信に対して適用した場合に発生する競合を，提案手法は全て回避した．実環境において，提案手法は既存手法と比較して多対 1 通信の場合に最も実行時間を削減し，6.3 倍の高速化を得た．一方，全対全通信に近い一部の通信パターンに対して実行時間が増加し，その増加率は最大 1.2 倍であった．提案手法は MPI 実装に対して，全対全通信の場合に最も実行時間を削減し，3.0 倍の高速化を得た．

今後の課題は，スループット低下の回避を目的としたスケジューリング手法の考案である．スループット低下の回避のためには，全ての競合を回避する必要はない．したがって，多対多通信の最適化のためには，スループット低下を回避する前提のもとで許容できる競合の発生度合を計算する必要がある．

謝辞 本研究の一部は，JSPS 科研費 JP15H01687 および JP16H02801 の補助による．

ⓒ 2020 Information Processing Society of Japan

参考文献

[1] Message Passing Interface Forum: MPI Documents, `https://www.mpi-forum.org/docs` (Accessed 2020-11-17).
[2] Wonnacott, D. G. and Strout, M. M.: On the Scalability of Loop Tiling Techniques, *In the Proceedings of 3rd International Workshop on Polyhedral Compilation (IMPACT '13)*, pp. 3–11 (2013).
[3] Demmel, J., Hoemmen, M., Mohiyuddin, M. and Yelick, K.: Avoiding communication in sparse matrix computations, *In the Proceedings of 2008 IEEE International Symposium on Parallel and Distributed Processing (PDP '08)*, IEEE, pp. 1–12 (2008).
[4] Okuyama, T., Okita, M., Abe, T., Asai, Y., Kitano, H., Nomura, T. and Hagihara, K.: Accelerating ODE-based Simulation of General and Heterogeneous Biophysical Models using a GPU, *IEEE Transactions on Parallel and Distributed Systems*, Vol. 25, No. 8, pp. 1966–1975 (2014).
[5] Thakur, R., Rabenseifner, R. and Gropp, W.: Optimization of collective communication operations in MPICH, *International Journal of High Performance Computing Applications*, Vol. 19, No. 1, pp. 49–66 (2005).
[6] Karwande, A., Yuan, X. and Lowenthal, D. K.: CC-MPI: a compiled communication capable MPI prototype for ethernet switched clusters, *ACM SIGPLAN Notices*, Vol. 38, No. 10, pp. 95–106 (2003).
[7] Alexandrov, A., Ionescu, M. F., Schauser, K. E. and Scheiman, C.: LogGP: Incorporating Long Messages into the LogP Model for Parallel Computation, *Parallel and Distributed Computing*, Vol. 44, No. 1, pp. 71 – 79 (1997).
[8] Pješivac-Grbović, J., Angskun, T., Bosilca, G., Fagg, G. E., Gabriel, E. and Dongarra, J. J.: Performance analysis of MPI collective operations, *Cluster Computing*, Vol. 10, No. 2, pp. 127–143 (2007).
[9] Barabási, A.-L. and Albert, R.: Emergence of Scaling in Random Networks, *Science*, Vol. 286, No. 5439, pp. 509–512 (1999).

ⓒ 2020 Information Processing Society of Japan