Vol.2020-HPC-175 No.11
2020/7/30

ノード間光ハブ接続を用いた並列行列積計算の高速化

賣野豊 [1]

概要：並列行列積計算（SUMMA）に対して，光ハブとそれに適した通信アルゴリズムの適用を提案し，その高速化効果を理論およびシミュレーションで検証した．その結果，最も高速な通信アルゴリズムでは，標準 MPI を用いた場合に比べて計算ノード数倍以上の高速化を確認した．また，同じ通信アルゴリズムおよびネットワーク仕様（総帯域幅，リンク遅延時間等）でネットワーク・トポロジーの違いによる比較を行うと，光ハブは 2D-torus に比べて約 7 倍，Fat-tree に比べて約 1.5 倍高速であることを確認した．

キーワード：並列計算，行列積計算，ネットワーク・トポロジー，通信アルゴリズム

Performance Enhancement for Matrix Multiplication on Parallel Computing Systems Connected with Optical Hub

YUTAKA URINO[†1]

1. はじめに

近年，処理すべきデータ量は指数関数的に増大する一方で，CPU の演算性能の伸びは鈍化しており，そのギャップを埋める手段としては主に並列化に頼っている[1]．分散メモリ型並列計算システムにおいては，計算ノード間の通信性能が十分でないと，それがシステム全体の性能のボトルネックとなり得る．我々はこれまでこの問題に対して，シリコンフォトニクスを用いた高帯域密度光トランシーバである「光 I/O コア」およびそれを用いて FPGA ボード間を接続したクラスタ・システムを開発してきた[2]．また，ノード間インターコネクトのトポロジーやルーティングに起因する問題も解決するため，波長ルーティングを用いたインターコネクトである「光ハブ」およびそのフルメッシュ論理トポロジーに適したルーティング・アルゴリズムである「マルチパス・ルーティング」を提案し，並列計算シミュレータを用いて並列計算ベンチマークの実行時間・演算時間・通信時間を解析し，アプリケーションの実行時間および並列計算システムの省エネの観点で，光ハブの優位性を示した[3][4].

行列積計算は，最も基本的な線形代数演算の一つであり，様々なアプリケーションの中で用いられる応用範囲の広い計算であるが，その計算量が概ね行列サイズの 3 乗に比例するため，多くのアプリケーションでその実行時間の大部分を占める場合が多く，行列積計算の高速化は重要である．特に，行列サイズが大きい場合は，行列を複数の小行列に分割して複数の計算ノードに分散配置し，計算ノード内での小行列同士の並列計算と計算ノード間の小行列の通信を繰り返して行列積を求める並列行列積計算が必要であり，その場合実行時間がノード間の通信性能に律速され易い傾向がある．

本論文では，並列行列積計算の代表的なアルゴリズムである SUMMA（Scalable Universal Matrix Multiplication Algorithm）[5]に対して，ノード間ネットワークには光ハブを，通信アルゴリズムにはマルチパス・ルーティングおよび更に通信回数を削減したアルゴリズムを適用することにより，通信時間および実行時間を短縮し，並列行列積計算を高速化することを検討する．本論文の構成は以下のとおりである．第 2 章では，これまでに提案した光ハブおよびマルチパス・ルーティングについて要点を説明する．第 3 章では，並列行列積計算のアルゴリズムとして SUMMA を取り上げ，その通信アルゴリズムの高速化を理論的に検討する．また必要となるメモリ容量についても検討する．第 4 章では，並列計算シミュレータを用いて，3 章で提案した通信アルゴリズムの高速化を検証する．第 5 章では，通信時間と演算時間の関係，および光ハブを行列積計算に適用したときのスケーラビリティ等について考察する．第 6 章は全体のまとめである．

2. ノード間インターコネクト「光ハブ」

2.1 構成と動作原理

光ハブによる 8 台の計算ノード（$N = 8$）間接続の物理トポロジーを図 1(a)に示す．波長ルータをハブとしたスター型の接続となる．簡単化のため，各ノードと波長ルータ間の接続は 1 本の線で表しているが，実際には双方向の通信を行うため，2 本の光ファイバで接続されている．例えば，送信元ノード#7 では，複数の送信先ノード#0 ～ #6 に対応したそれぞれ波長λ_1～λ_7のキャリア光を変調し，これら複数の波長を波長多重した上で 1 本の光ファイバ経由で波

1 技術研究組合 光電子融合基盤技術研究所
Photonics Electronics Technology Research Association (PETRA)

長ルータに送信する．波長ルータのある入力ポートに入力したある波長の信号は，図 1(b)に示す周期的なルーティング・テーブルによって決まる出力ポートに出力される[6][7]．例えば，ノード#7 に接続された入力ポートから入力した波長λ_4の信号は，ノード#3 に接続された出力ポートに出力される．送信先（受信）ノードでは，1 本の光ファイバ経由で入力した複数の波長を分波し，送信元ノードに対応した受信機で受信する．

図 1(b)の表の各セルはグラフ理論の隣接行列の各要素に対応する．本論文では，この各セルに対応する送受信ノード間の（一方向の）接続をリンクと呼ぶ．光ハブは，この表の全てのセルに対応した 64 本（N^2本）リンクの内対角項を除いた 56 本（$N(N-1)$本）のリンクを有し，その全てのリンクで同時に通信を実行できる．したがって，その論理トポロジーは図 2 に示す通りフルメッシュ（グラフ理論では完全グラフ）となる．図 2 でノード間の 1 本の線は，双方向のリンクを表している．

各計算ノードのネットワーク・インターフェースの構成の詳細は紙面の都合で文献[8][9]に譲るが，比較的低速な（ノード数−1）台の高密度光トランシーバと，その光トランシーバを選択する経路スイッチ（分散スイッチと呼ぶ）を有することが特徴である．我々は上記光トランシーバを光 I/O コアと同様のシリコンフォトニクス技術で，分散スイッチを FPGA で実現しようとしている[3][4]．

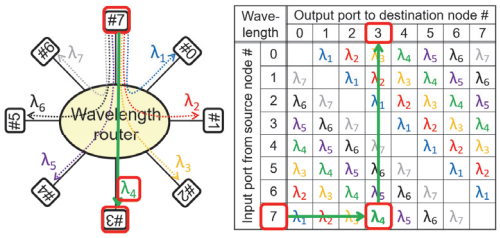

図 1　光ハブの物理トポロジー(a)と
波長ルータのルーティング・テーブル(b)

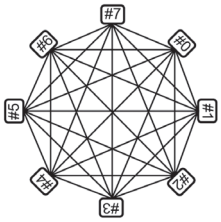

図 2　光ハブの論理トポロジー

2.2 特徴と優位性

光ハブの特徴を従来の Fat-tree 型パケットスイッチ・ネットワークとの比較として表 1 に示す．詳細な説明は文献[3][4]に譲るが，光ハブは従来の Fat-tree 型パケットスイッチ・ネットワークに比べて，低消費電力，高帯域密度，低遅延，高信頼性等の特徴がある．

表 1　光ハブと従来型 Fat-tree ネットワークの比較

項目		従来型 Fat-tree	光ハブ
ハブ	ルーティング方式	電気スイッチ（アクティブ）	波長ルータ（パッシブ）
	消費電力	スループット比例	スループット無依存 **⇒ 低消費電力**
	ポート数制限要因（上限数）	電気コネクタ密度（40）	波長数（80） **⇒ 高帯域密度**
リンク	多重化	時分割（パケット）	波長多重（常時接続）
	パケット処理	複雑	簡単 **⇒ 低遅延**
	リンク障害対応	迂回困難	迂回容易 **⇒ 高信頼性**

2.3 マルチパス・ルーティング

第 3 章で詳しく述べるが，SUMMA ではノード間の行列データの通信にブロードキャスト（以下，Bcast）通信が繰り返し使われる．そこでこの節では，論理的フルメッシュ接続で，Bcast 通信等を高速化するマルチパス・ルーティング[3][4]について説明する．ノード数$N=4$の場合を例に，通信パスとして送受信ノード間のリンクだけを使う（直接ルーティングと呼ぶ）Bcast 通信を図 3(a)に示す．この場合，フルメッシュ接続の$N(N-1)$本の総リンクの内，$N-1$本しか使われず，リンクの利用効率が低い．一方，マルチパス・ルーティングによる Bcast 通信を図 3(b)に示す．送信（ルート）ノード内で送信データはN分割され，まずその内の$N-1$個が送信ノード以外の$N-1$ノード（中継ノードと呼ぶ）に Scatter され，次に送信ノードに残った 1 個と$N-1$個の中継ノードのデータが各受信ノードへ Allgather される．この方法により，1 リンク当たりのメッセージサイズは$1/N$に，通信回数は 2 倍になるので，実効帯域幅を直接ルーティングに対して，$N/2$倍に拡大することができる．したがって，光ハブとマルチパス・ルーティングを用いることで SUMMA の高速化が期待できる．なお，他の通信パターン（Send/Recv, Reduce, Allreduce 等）についても，Bcast と同様にマルチパス・ルーティングにより実効帯域幅の$N/2$倍化が可能である．

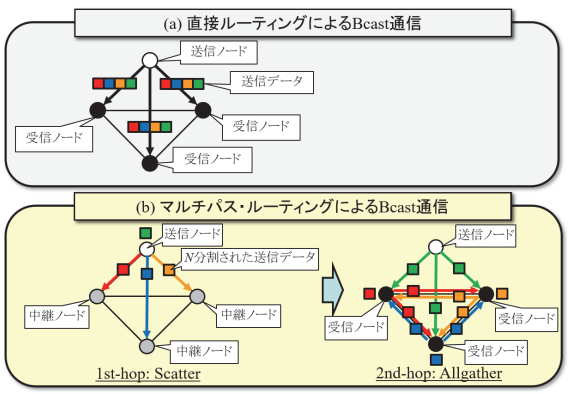

図 3　直接ルーティングとマルチパス・ルーティング

Vol.2020-HPC-175 No.11
2020/7/30

3. 並列行列積計算の通信アルゴリズム

3.1 従来のアルゴリズム

並列行列積計算の代表的な通信アルゴリズムを表 2 に示す．Cannon[10]および Fox[11]の方法は，行列データの計算ノード間の通信に循環シフト型の通信が用いられるため，ネットワーク・トポロジーが 2 次元トーラス等の場合に比較的効率良く通信が行える特徴があるが，計算ノードの数が 2 乗数に限定される制約がある．一方 SUMMA は，行列データの計算ノード間の通信に Bcast 通信が用いられるため，Bcast 通信を効率よく行えるネットワークに適しており，また上記の計算ノード数に対する制限が無いことが特徴である[5]．2.3 節で示した通り，光ハブではマルチパス・ルーティングを用いることで Bcast 通信を効率良く行うことが出来るので，本論文では SUMMA を基本のアルゴリズムとし，その高速化を検討した．

表 2　従来の並列行列積計算の通信アルゴリズム

アルゴリズム	Cannon	Fox	SUMMA
行列 **A** の通信	循環シフト	循環シフト	ブロードキャスト
行列 **B** の通信	ブロードキャスト	循環シフト	ブロードキャスト
計算ノード数の制限	2 乗数のみ	2 乗数のみ	制限無し
参考文献	[10]	[11]	[5]

例として，9 台（$N = 9$）の計算ノード（P_0 ～ P_8）を用いて行列 **A** と **B** の行列積 **C**=**AB** を並列計算する SUMMA のアルゴリズムを図 4 に示す．(a)計算ノードは仮想的に 3×3（$\sqrt{N} \times \sqrt{N}$）の 2 次元グリッド状に構成される．(b)行列 **A**, **B** および **C** はそれぞれ 3×3 の 9 つの小行列（$\mathbf{A}_{11}$ ～ $\mathbf{A}_{33}$, $\mathbf{B}_{11}$ ～ $\mathbf{B}_{33}$ および $\mathbf{C}_{11}$ ～ $\mathbf{C}_{33}$）に分割され，それぞれ対応する計算ノード（P_0 ～ P_8）に配置される．**C** の各小行列の各要素の初期値は 0 である．(c)～(e)並列行列積の計算は，3 回（$\sqrt{N}$回）のノード間通信とノード内の小行列同士の計算のセットによって行われる．k 回目（k=1, 2, 3）の通信と計算のセットでは，k 列目の計算ノードに配置された小行列 $\mathbf{A}_{ik}$（i=1, 2, 3）を同じ行に属するノードに対して Bcast し，k 行目の計算ノードに配置された小行列 $\mathbf{B}_{kj}$（j=1, 2, 3）を同じ列に属するノードに対して Bcast し，各ノードでは Bcast された小行列 $\mathbf{A}_{ik}$ と $\mathbf{B}_{kj}$ の行列積 $\mathbf{A}_{ik}\mathbf{B}_{kj}$ を計算し，$\mathbf{C}_{ij}$ に加算する．

このアルゴリズム（以下，CA1 と呼ぶ）の 3 回の通信パターンを図 5(a) ～ (c)に示す．図 5 の表の各セルは図 1(b)と同様にグラフ理論の隣接行列の各要素に対応する．CA1 のアルゴリズムでは，各セットの Bcast 通信で利用されるリンクは全リンク 72 本の内 12 本に過ぎず，リンクの利用効率が低い．図 5(a) ～ (c)の 3 回の Bcast 通信では，同じリンクが複数回使われることはない．したがって，図 5(d)に示すように 3 回の Bcast 通信を 1 回にまとめることでリンクの利用効率を上げ（通信回数減らし）通信時間を短縮することができる．このアルゴリズムを CA2 と呼ぶ．

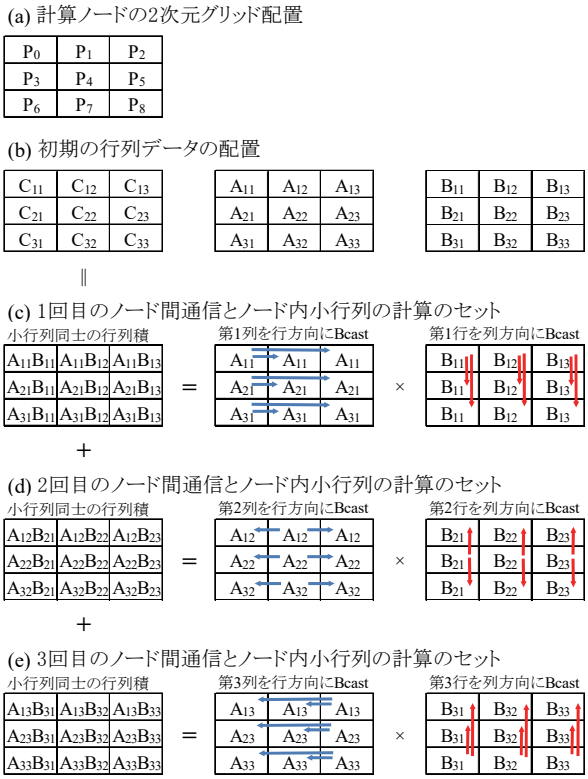

図 4　従来の SUMMA のアルゴリズム(CA1)

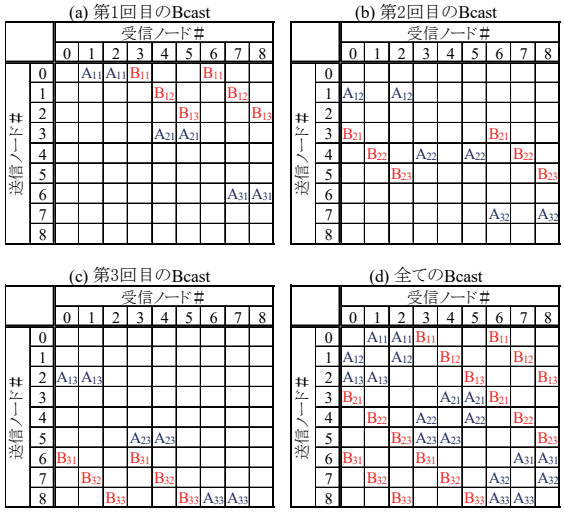

図 5　SUMMA のノード間通信パターン
(a) ～ (c)：従来のアルゴリズム(CA1),
(d)：Bcast を 1 回にしたアルゴリズム(CA2)

3.2 マルチパス・ルーティングによる高速化

マルチパス・ルーティングを用いることで，更に通信時間を短縮することが出来る．図 6 は 3 回の Bcast 通信をマルチパス･ルーティングにした場合の通信パターンを示す．紙面の都合で行列 **A** のみ示すが，行列 **B** に関してもほぼ同様である．ここで $\mathbf{A}_{ijk}$ は小行列 $\mathbf{A}_{ij}$ を 9 分割した k 番目（k=0, … , 8）の小行列である．各 Bcast は Scatter と Allgather の組に置き換えらえるため，通信回数は 2 倍に増えるが，各

メッセージサイズが 1/9 になるため，通信時間が帯域律速であれば通信時間を短縮できる．このアルゴリズムを CA3 と呼ぶ．

図 6(a), (c), (e)に着目すると，3 回の Scatter で同じリンクは使われていないので，これらを 1 回の Alltoall にまとめることが出来る．また(b), (d), (f)の Allgather についても，例えば 3 回目(f)の Allgather の半分を 1 回目(b)，もう半分を 2 回目(d)の Allgather と同時に行うことで，通信回数を 1 回分削減出来る．それらを適用した通信パターンを図 7 に示す．これにより，全ての通信において，全てのリンクを使う通信パターンとなり，これ以上リンクの利用効率を上げることは出来ない．このアルゴリズムを CA4 と呼ぶ．

(a) 第1回目のScatter

送信ノード＃ \ 受信ノード＃	0	1	2	3	4	5	6	7	8
0	A_{110}	A_{111}	A_{112}	A_{113}	A_{114}	A_{115}	A_{116}	A_{117}	A_{118}
1									
2									
3	A_{210}	A_{211}	A_{212}	A_{213}	A_{214}	A_{215}	A_{216}	A_{217}	A_{218}
4									
5									
6	A_{310}	A_{311}	A_{312}	A_{313}	A_{314}	A_{315}	A_{316}	A_{317}	A_{318}
7									
8									

(b) 第1回目のAllgather

送信ノード＃ \ 受信ノード＃	0	1	2	3	4	5	6	7	8
0		A_{110}	A_{110}		A_{210}	A_{210}		A_{310}	A_{310}
1		A_{111}	A_{111}		A_{211}	A_{211}		A_{311}	A_{311}
2		A_{112}	A_{112}		A_{212}	A_{212}		A_{312}	A_{312}
3		A_{113}	A_{113}		A_{213}	A_{213}		A_{313}	A_{313}
4		A_{114}	A_{114}		A_{214}	A_{214}		A_{314}	A_{314}
5		A_{115}	A_{115}		A_{215}	A_{215}		A_{315}	A_{315}
6		A_{116}	A_{116}		A_{216}	A_{216}		A_{316}	A_{316}
7		A_{117}	A_{117}		A_{217}	A_{217}		A_{317}	A_{317}
8		A_{118}	A_{118}		A_{218}	A_{218}		A_{318}	A_{318}

(c) 第2回目のScatter

送信ノード＃ \ 受信ノード＃	0	1	2	3	4	5	6	7	8
0									
1	A_{120}	A_{121}	A_{122}	A_{123}	A_{124}	A_{125}	A_{126}	A_{127}	A_{128}
2									
3									
4	A_{220}	A_{221}	A_{222}	A_{223}	A_{224}	A_{225}	A_{226}	A_{227}	A_{228}
5									
6									
7	A_{320}	A_{321}	A_{322}	A_{323}	A_{324}	A_{325}	A_{326}	A_{327}	A_{328}
8									

(d) 第2回目のAllgather

送信ノード＃ \ 受信ノード＃	0	1	2	3	4	5	6	7	8
0	A_{120}		A_{120}	A_{220}		A_{220}	A_{320}		A_{320}
1	A_{121}		A_{121}	A_{221}		A_{221}	A_{321}		A_{321}
2	A_{122}		A_{122}	A_{222}		A_{222}	A_{322}		A_{322}
3	A_{123}		A_{123}	A_{223}		A_{223}	A_{323}		A_{323}
4	A_{124}		A_{124}	A_{224}		A_{224}	A_{324}		A_{324}
5	A_{125}		A_{125}	A_{225}		A_{225}	A_{325}		A_{325}
6	A_{126}		A_{126}	A_{226}		A_{226}	A_{326}		A_{326}
7	A_{127}		A_{127}	A_{227}		A_{227}	A_{327}		A_{327}
8	A_{128}		A_{128}	A_{228}		A_{228}	A_{328}		A_{328}

(e) 第3回目のScatter

送信ノード＃ \ 受信ノード＃	0	1	2	3	4	5	6	7	8
0									
1									
2	A_{130}	A_{131}	A_{132}	A_{133}	A_{134}	A_{135}	A_{136}	A_{137}	A_{138}
3									
4									
5	A_{230}	A_{231}	A_{232}	A_{233}	A_{234}	A_{235}	A_{236}	A_{237}	A_{238}
6									
7									
8	A_{330}	A_{331}	A_{332}	A_{333}	A_{334}	A_{335}	A_{336}	A_{337}	A_{338}

(f) 第3回目のAllgather

送信ノード＃ \ 受信ノード＃	0	1	2	3	4	5	6	7	8
0	A_{130}	A_{130}		A_{230}	A_{230}		A_{330}	A_{330}	
1	A_{131}	A_{131}		A_{231}	A_{231}		A_{331}	A_{331}	
2	A_{132}	A_{132}		A_{232}	A_{232}		A_{332}	A_{332}	
3	A_{133}	A_{133}		A_{233}	A_{233}		A_{333}	A_{333}	
4	A_{134}	A_{134}		A_{234}	A_{234}		A_{334}	A_{334}	
5	A_{135}	A_{135}		A_{235}	A_{235}		A_{335}	A_{335}	
6	A_{136}	A_{136}		A_{236}	A_{236}		A_{336}	A_{336}	
7	A_{137}	A_{137}		A_{237}	A_{237}		A_{337}	A_{337}	
8	A_{138}	A_{138}		A_{238}	A_{238}		A_{338}	A_{338}	

図 6　マルチパス・ルーティングを用いたアルゴリズム(CA3)の通信パターン

(a) 第1回目のAlltoall

送信ノード＃ \ 受信ノード＃	0	1	2	3	4	5	6	7	8
0	A_{110}	A_{111}	A_{112}	A_{113}	A_{114}	A_{115}	A_{116}	A_{117}	A_{118}
1	A_{120}	A_{121}	A_{122}	A_{123}	A_{124}	A_{125}	A_{126}	A_{127}	A_{128}
2	A_{130}	A_{131}	A_{132}	A_{133}	A_{134}	A_{135}	A_{136}	A_{137}	A_{138}
3	A_{210}	A_{211}	A_{212}	A_{213}	A_{214}	A_{215}	A_{216}	A_{217}	A_{218}
4	A_{220}	A_{221}	A_{222}	A_{223}	A_{224}	A_{225}	A_{226}	A_{227}	A_{228}
5	A_{230}	A_{231}	A_{232}	A_{233}	A_{234}	A_{235}	A_{236}	A_{237}	A_{238}
6	A_{310}	A_{311}	A_{312}	A_{313}	A_{314}	A_{315}	A_{316}	A_{317}	A_{318}
7	A_{320}	A_{321}	A_{322}	A_{323}	A_{324}	A_{325}	A_{326}	A_{327}	A_{328}
8	A_{330}	A_{331}	A_{332}	A_{333}	A_{334}	A_{335}	A_{336}	A_{337}	A_{338}

(b) 第1回目のAlltoallv

送信ノード＃ \ 受信ノード＃	0	1	2	3	4	5	6	7	8
0	A_{130}	A_{110}	A_{110}	A_{230}	A_{210}	A_{210}	A_{330}	A_{310}	A_{310}
1	A_{131}	A_{111}	A_{111}	A_{231}	A_{211}	A_{211}	A_{331}	A_{311}	A_{311}
2	A_{132}	A_{112}	A_{112}	A_{232}	A_{212}	A_{212}	A_{332}	A_{312}	A_{312}
3	A_{133}	A_{113}	A_{113}	A_{233}	A_{213}	A_{213}	A_{333}	A_{313}	A_{313}
4	A_{134}	A_{114}	A_{114}	A_{234}	A_{214}	A_{214}	A_{334}	A_{314}	A_{314}
5	A_{135}	A_{115}	A_{115}	A_{235}	A_{215}	A_{215}	A_{335}	A_{315}	A_{315}
6	A_{136}	A_{116}	A_{116}	A_{236}	A_{216}	A_{216}	A_{336}	A_{316}	A_{316}
7	A_{137}	A_{117}	A_{117}	A_{237}	A_{217}	A_{217}	A_{337}	A_{317}	A_{317}
8	A_{138}	A_{118}	A_{118}	A_{238}	A_{218}	A_{218}	A_{338}	A_{318}	A_{318}

(c) 第2回目のAlltoallv

送信ノード＃ \ 受信ノード＃	0	1	2	3	4	5	6	7	8
0	A_{120}	A_{130}	A_{120}	A_{220}	A_{230}	A_{220}	A_{320}	A_{330}	A_{320}
1	A_{121}	A_{131}	A_{121}	A_{221}	A_{231}	A_{221}	A_{321}	A_{331}	A_{321}
2	A_{122}	A_{132}	A_{122}	A_{222}	A_{232}	A_{222}	A_{322}	A_{332}	A_{322}
3	A_{123}	A_{133}	A_{123}	A_{223}	A_{233}	A_{223}	A_{323}	A_{333}	A_{323}
4	A_{124}	A_{134}	A_{124}	A_{224}	A_{234}	A_{224}	A_{324}	A_{334}	A_{324}
5	A_{125}	A_{135}	A_{125}	A_{225}	A_{235}	A_{225}	A_{325}	A_{335}	A_{325}
6	A_{126}	A_{136}	A_{126}	A_{226}	A_{236}	A_{226}	A_{326}	A_{336}	A_{326}
7	A_{127}	A_{137}	A_{127}	A_{227}	A_{237}	A_{227}	A_{327}	A_{337}	A_{327}
8	A_{128}	A_{138}	A_{128}	A_{228}	A_{238}	A_{228}	A_{328}	A_{338}	A_{328}

図 7　更に通信回数を削減したアルゴリズム(CA4)の通信パターン

3.3 2.5 次元マッピングによる高速化

並列行列積計算で通信回数を減らす別の方法として，例えば図 4 で 9 台の 2 次元グリッド状の計算ノードを使って 3 セットの通信を行っていた（空間 2 次元＋時間 1 次元の）アルゴリズムを，3 層のコピー層からなる 27 台の 3 次元グリッド状の計算ノードを用いて空間 3 次元に割り当て（マッピングし），行列 **A**, **B** に対してはそれぞれ Bcast 通信を 1 回に削減し，　**C** に対しては Reduce 通信を 1 回行う方法が知られている[12]．この方法を 2.5 次元マッピングと呼ぶ．ただしこの場合，ノード数を一定で考えると，1 ノードに割り振られる小行列のデータ長は，前節までの方法（2 次元マッピングと呼ぶ）に比べて$N^{1/3}$倍大きくなるため，1 ノード当たりの計算量や必要となるメモリ容量の点では不利になる．2.5 次元マッピングの場合でも，その Bcast および Reduce 通信にマルチパス・ルーティングによる CA3 と同様のアルゴリズムの併用が可能である．

3.4 各アルゴリズムの通信時間の見積

これまで示した各通信アルゴリズムによる通信時間の理論値の見積を行った．通信時間t_cは，式(1)で表される．

$$t_c = M(\frac{s}{b} + L) \qquad (1)$$

ここで，sはメッセージサイズ，bはリンク帯域幅，Mは通信回数，Lはリンク遅延時間である．今回は比較的大きな行列同士の行列積を想定するので，式(1)の右辺括弧内の第 2 項は第 1 項に比べて無視できる，すなわち通信時間は帯域律速であると仮定した．

表 3 に各アルゴリズムの通信時間を比較した．ここで，Nはノード数，nは行列サイズであり，1 ノードの総帯域幅をBとした．本来，データ長sおよび通信時間t_cには行列データのビット長（例えば倍精度なら 64 bits）が掛かるが，全てに共通のため省略した．光ハブは論理フルメッシュであり，ノード数Nの場合，1 ノード当たりN本のリンクを持つので（厳密には$N-1$本であるが，Nがある程度大きいことを想定し，簡素化のためNで近似する），1 リンク当たりの帯域幅$b = B/N$となる．2.5 次元マッピングの行列の初期データは，$N^{1/3} \times N^{1/3}$の 2 次元グリッド状の計算ノードに配置されており，$N^{1/3}$層のコピー層に Bcast されるものとした．

通信時間t_cの逆数$1/t_c$を相対性能（CA1 の値で規格化）と定義すると，2 次元マッピングの CA1 から CA4 の相対性能は，それぞれ，$1, 2\sqrt{N}, n/2, N$となり，CA1→CA4 の順に高速化され，CA4 が最も通信時間が短縮されると見込まれる．2 次元マッピングと 2.5 次元マッピングを比較すると，ノード数$N > 3^6 = 729$の場合は，（2.5.次元マッピング，CA3）の組み合わせが最も通信時間が短くなるが，$N < 729$では（2 次元マッピング，CA4）の組み合わせが最も通信時間が短くなると見込まれる．

一般に，従来のネットワークでは，Bcast 通信時間は少なくともノード数Nに対して$\log_2 N$に比例して増えるが，表 3 に示す

通り，SUMMA に光ハブと（2 次元マッピング，CA4）または（2.5 次元マッピング，CA3）を適用することにより，通信時間をそれぞれ$1/\sqrt{N}$または$1/N^{2/3}$に比例して減少させられることは注目すべき点だと考える．

表 3　SUMMA の各通信アルゴリズムの比較

行列マッピング次元	アルゴリズム	リンク帯域幅 b	通信回数 M	データ長 s	通信時間 $t_c = Ms/b$	相対理論性能	
						$1/t_c$	$N = 64$
2	CA1	B/N	$2\sqrt{N}$	n^2/N	$2n^2\sqrt{N}/B$	1	1.0
2	CA2	B/N	1	n^2/N	n^2/B	$2\sqrt{N}$	16.0
2	CA3	B/N	$4\sqrt{N}$	$(n/N)^2$	$4n^2/(\sqrt{N}B)$	$N/2$	32.0
2	CA4	B/N	$2\sqrt{N}$	$(n/N)^2$	$2n^2/(\sqrt{N}B)$	N	64.0
2.5	CA1	B/N	3	$n^2/N^{\frac{2}{3}}$	$3n^2N^{\frac{1}{3}}/B$	$2N^{\frac{1}{6}}/3$	1.3
2.5	CA3	B/N	6	$n^2/N^{\frac{5}{3}}$	$6n^2/(N^{\frac{2}{3}}B)$	$N^{\frac{7}{6}}/3$	42.7

3.5 必要となるメモリ容量

SUMMA で必要となる 1 ノード当たりのメモリ容量を見積もった．行列 **A**, **B**, **C** のサイズが$n \times n$で，これをN台の計算ノードに 2 次元マッピングした場合，1 ノード当たりの各小行列のデータ長はそれぞれn^2/Nである（ここでもデータのビット長は省略した）．CA1 アルゴリズムの場合は，小行列 $\mathbf{A}_{ij}, \mathbf{B}_{ij}, \mathbf{C}_{ij}$ と小行列 $\mathbf{A}_{ij}, \mathbf{B}_{ij}$ の Bcast 時のバッファが必要になるため，総メモリ容量は，$5n^2/N$となる．同様に，他の通信アルゴリズムおよび行列マッピングの場合のメモリ容量を見積もり，表 4 に示す．

（2 次元マッピング，CA2）の組み合わせおよび 2.5 次元マッピングは，他の方法に比べて必要とするメモリ容量が大きく，その傾向はNが大きいほど顕著になる．その理由は，前者は CA1 等では$\sqrt{N}$回に分けて行われる Bcast を 1 回で済ませるため，$\sqrt{N}$倍の Bcast 用バッファメモリが必要になるためである．また後者は，$N^{1/3}$層のコピー・ノードを設けるため，1 ノード当たりの小行列のデータ長が 2 次元マッピングに比べ$N^{1/3}$倍になることに加え，小行列 $\mathbf{C}_{ij}$ の Reduce 通信用のバッファも必要になるためである．

表 4　SUMMA の必要メモリ容量の比較

行列マッピング	アルゴリズム	必要メモリ容量		
		理論値	相対値	$N = 64$
2 次元	CA1	$5n^2/N$	1	1.0
2 次元	CA2	$(3 + 2\sqrt{N})n^2/N$	$(3 + 2\sqrt{N})/5$	3.8
2 次元	CA3	$(5 + 2/\sqrt{N})n^2/N$	$1 + 2/(5\sqrt{N})$	1.05
2 次元	CA4	$7n^2/N$	7/5	1.4
2.5 次元	CA1	$6n^2/N^{\frac{2}{3}}$	$6N^{\frac{1}{3}}/5$	4.8
2.5 次元	CA3	$6n^2/N^{\frac{2}{3}} + 2n^2/N$	$(2 + 6N^{\frac{1}{3}})/5$	5.2

表 3 に示した通信時間の逆数（相対性能）と表 4 に示した必要メモリ容量の比を表 5 に示す．簡単化のため，必要メモリ容量の理論値はNが大きい場合の漸近値とした．2 次元マッピングで CA3 または CA4 の通信アルゴリズム用いた場合，その他の場合に比べて，必要メモリ容量当たりの相対性能が突出して高く，その傾向はNが大きくなるほどより顕著になることが分かる．

表 5　相対性能／必要メモリ容量の比較

行列マッピング	アルゴリズム	相対性能	必要メモリ容量	相対性能／必要メモリ容量	
		理論値	理論値(漸近値)	理論値	相対値 $(N = 64)$
2 次元	CA1	$B/(2n^2\sqrt{N})$	$5n^2/N$	$BN^{\frac{1}{2}}/(10n^4)$	1.0
2 次元	CA2	B/n^2	$2\sqrt{N}n^2/N$	$BN^{\frac{1}{2}}/(2n^4)$	4.2
2 次元	CA3	$\sqrt{N}B/(4n^2)$	$5n^2/N$	$BN^{\frac{3}{2}}/(20n^4)$	30.5
2 次元	CA4	$\sqrt{N}B/(2n^2)$	$7n^2/N$	$BN^{\frac{3}{2}}/(14n^4)$	45.7
2.5 次元	CA1	$B/(3n^2N^{\frac{1}{3}})$	$6n^2/N^{\frac{2}{3}}$	$BN^{\frac{1}{3}}/(18n^4)$	0.3
2.5 次元	CA3	$N^{\frac{2}{3}}B/(6n^2)$	$6n^2/N^{\frac{2}{3}}$	$BN^{\frac{4}{3}}/(36n^4)$	8.2

4. 性能評価シミュレーション

4.1 並列計算シミュレータ SimGrid

3 章で提案した通信アルゴリズムの性能を検証するため，並列計算シミュレータ SimGrid によるシミュレーションを行った．SimGrid は既存の MPI を使った並列計算コードを仮想並列計算プラットフォーム上で実行し，その実行時間をシミュレーションすることができる[13].

4.2 仮想プラットフォームの仕様

SimGrid を用いてシミュレーションを行う際の仮想プラットフォーム（ハードウエア）の仕様を表 6 に示す．ネットワーク・トポロジーとして，光ハブおよび比較のため 2D-torus および 2-layer fat-tree を用いた．計算ノード数Nは 64，計算ノードの演算能力Pは 1 TFlops，計算ノードのネットワークの総帯域幅Bは 1600 Gbps （200 GByte/s），リンク遅延時間Lは 100 ns で共通とした．各計算ノードは，2D-torus は 4，Fat-tree は 1，光ハブは 64 のネットワークポートを持つので，1 ポート当たりの帯域幅bは，それぞれ 400, 1600, 25 Gbps となる．2-layer fat-tree は，1600 Gbps×32 ports のパケットスイッチ 6 台（1 層目に 4 台，2 層目に 2 台）で構成され，1 層目の各スイッチに 16 台の計算ノードが接続される．

表 6　仮想プラットフォームの仕様

ネットワーク・トポロジー	2D-torus	2-layer fat-tree	光ハブ
ノード数 N	64	64	64
1ノードの演算能力 P (TFlops)	1	1	1
1ノードの総帯域幅 B (Gbps)	1600	1600	1600
1ノードのポート数	4	1	64
1ポートの帯域幅 b (Gbps)	400	1600	25
リンク遅延時間 L (ns)	100	100	100

4.3 集合通信時間

SimGrid でシミュレーションした Bcast の通信時間を図 8 に示す．破線および一点鎖線は，それぞれ光ハブの直接ルーティングおよびマルチパス・ルーティングの理論値を表しており，光ハブのシミュレーション結果は概ね理論値と一致することが確認できた．今回設定の帯域幅および遅延時間では，メッセージサイズが 512 Byte より小さいときには直接ルーティング，それより大きいときにはマルチパス・ルーティングの方が高速になる．比較対象の 2D-torus および 2-layer fat-tree では，標準の MPI 実装として MVAPICH2 を用いた．図示は省略したが，Fat-tree のシミュレーション結果は，2 分木ルーティングの理論値と概ね一致していた．メッセージサイズが大きい領域では，光ハブは従来型ネットワークに比べて 1 桁程度高速である．

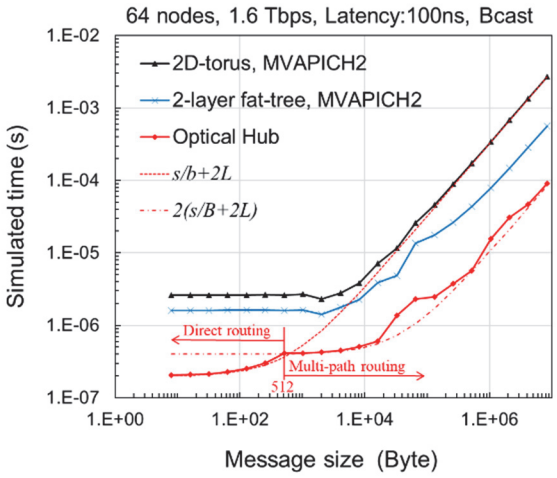

図 8　Bcast 通信の通信時間シミュレーション

4.4 並列行列積計算（SUMMA）の実行時間

ノード間ネットワークに光ハブ，行列マッピングに 2 次元マッピングを用いた場合の SUMMA の実行時間を図 9 に示す．各行列データは倍精度（64 bits）を用いた．マークと実線は SimGrid によるシミュレーション結果，破線は表 3 の通信時間t_cにビット長 64 bits を掛けた値である．ここで，CA0 は Bcast 通信に MPI_Bcast を用いた場合の実行時間である．本来ならばこの CA0 の実行時間が 3 章で示した CA1 の実行時間に相当するはずであるが，実際にはその約 2 倍の実行時間となった．これは，今回シミュレーションに用いた MPI 実装（MVAPICH2）の MPI_Bcast がフルメッシュ・ネットワークに対して最適化されていないためだと推定している．そこで CA1 の Bcast 通信では，MPI_Isend と MPI_Irecv を繰り返す方法を採用したところ，3 章の予測した実行時間とほぼ一致する結果が得られた．2 次元マッピングの Bcast のメッセージサイズは，横軸の左端$n = 128$のとき2 kByte，右端の$n = 8192$のとき8 MByteであるから，同図の横軸の全範囲で，直接ルーティングよりマルチパス・ルーティングの方が高速になる．通信アルゴリズムに依らず，行列サイズが大きい領域では，SimGrid の実行時間は表 3 に示した理論値に漸近し，実行時間が通信帯域律速であることが分かる．

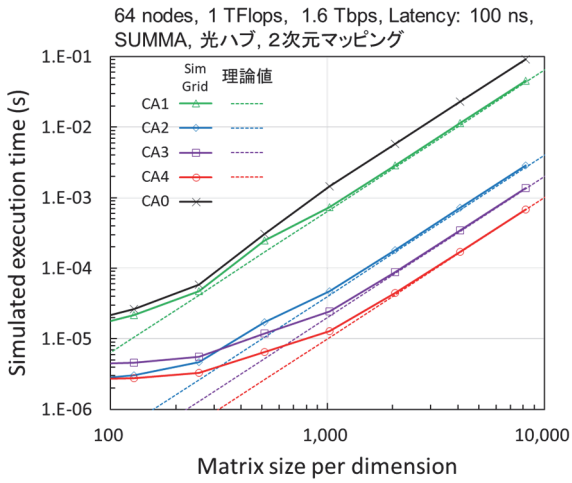

図 9　SUMMA 実行時間のシミュレーション

行列サイズが図 9 の右端（8192×9192）の場合の相対性能（実行時間の逆数，（2 次元マッピング，CA1，光ハブ）の値で規格化）を表 7 に示す．ネットワークを光ハブに固定し，通信アルゴリズム依存性を見ると，その相対性能は 3 章で示した理論予測とほぼ一致した値がシミュレーションによっても得られた．これは実行時間が通信時間律速であり，光ハブの論理フルメッシュ・トポロジーでは通信パケット同士の衝突が発生しないため，理論通信時間通りの実行時間がシミュレーションでも得られたためと推定している．CA4 アルゴリズムを用いることで，CA0 アルゴリズムに比べてノード数(64)倍以上に高速となった．

一方，通信アルゴリズムを CA4 に固定し，ネットワーク依存性を見ると，光ハブは 2D-torus に比べて 1 桁以上，Fat-tree に比べて 50%以上高速となった．

全ての組み合わせの中で比較すると，（2 次元マッピング，CA4，光ハブ）の組み合わせが最も高速であり，次が（2 次元マッピング，CA2，Fat-tree）であるが，前述の通り CA2 は CA4 に比べて大きなメモリ容量を必要とする．また，2.2 節で簡単に述べたが，システム全体の消費電力量，サイズ，信頼性等の点でも，光ハブには Fat-tree に比べて優位性がある[3][4]．例えば，現時点で利用可能な 1U サイズのパケットスイッチの最大容量は，40-port InfiniBand HDR(200 Gbps)スイッチ[14]の 8 Tbps 程度であると思われるが，4.2 節で述べた通り今回のシミュレーションでは 2-layer fat-tree 用としてスイッチ容量が 51.2 Tbps(1.6 Tbps×32 ports)のスイッチ 6 台を用いることを想定した．したがって，単純にスイッチ容量の比から換算すると，今回の 2-layer fat-

tree を実現するためには上記 1U スイッチが 38 台必要になり，更に各ノードには HDR(200 Gbps)ケーブルを 8 本接続する必要があり，システムが巨大になる．したがって，(2 次元マッピング，CA4，光ハブ）の組み合わせは，この表で示した相対性能の差以上に他の組み合わせに対して優位性があると考えている．

表 7　SUMMA 実行時間の比較
(ノード数：64，行列サイズ：8192×8192)

行列マッピング	アルゴリズム	相対性能（理論）	相対性能　（SimGrid）		
			光ハブ	2D-torus	2-layer fat-tree
2 次元	CA0	—	0.5	3.5	9.9
2 次元	CA1	1.0	1.0	5.8	9.6
2 次元	CA2	16.0	16.0	27.0	64.7
2 次元	CA3	32.0	33.5	4.5	26.5
2 次元	CA4	64.0	66.8	5.6	43.5
2.5 次元	CA1	1.3	1.3	3.1	15.0
2.5 次元	CA3	42.7	42.2	4.4	29.6

5. 考察

5.1 演算時間と通信時間

並列計算の高速化を検討する上で，演算時間と通信時間の比率を知ることは重要である．4 章のシミュレーションでは，通信時間が演算時間に比べて長く，実行時間の大部分を占めていたため，通信時間の短縮がほぼそのまま実行時間の短縮に繋がった．本節では，この条件の適用範囲を検討する．アプリケーションの実行時間tを演算時間t_pと通信時間t_cに分け，式(2)の通り仮定する．

$$t(P,B,L) = t_p + t_c, \quad t_p = \frac{C}{P}, \quad t_c = \frac{S}{B} + ML \qquad (2)$$

ここで，t_cは演算時間と重なる部分を除いた通信時間，Pは 1 ノードの演算能力(Flops)，Bは 1 ノードの総帯域幅 (bps または Byte/s), C, S, Mはそれぞれ 1 ノードの総演算量(Flop)，ノード間の総通信量(Byte)，および総通信回数である．SimGrid ではプラットフォームのパラメータであるP, B, Lを任意の値に設定できるので，これらを 2 種類ずつ設定してアプリケーションの実行時間$t(P,B,L)$をシミュレーションすることにより，アプリケーションのパラメータであるC, S, Mを逆算し，演算時間t_pおよび通信時間t_cを分離することが出来る[3][4]. 前述の通り，通信アルゴリズムが CA4 の場合の通信時間t_cは，

$$t_c \propto \frac{n^2}{B\sqrt{N}} \qquad (3)$$

である．一方，演算時間t_pは，

$$t_p \propto \frac{n^3}{PN} \qquad (4)$$

であるから，通信時間t_cと演算時間t_pの比は，

$$\frac{t_c}{t_p} \propto \frac{\sqrt{N}}{n}\frac{P}{B} \qquad (5)$$

となる．図 10 に（2 次元マッピング，CA4，光ハブ）を用い，SimGrid でシミュレーションした SUMMA の実行時間から，通信時間t_cおよび演算時間t_pを求め，その比をプロットした．このプロットから，通信アルゴリズム CA4 では式(5)が成り立っていることが確認できた．また，回帰式から通信時間が演算時間より長くなる，すなわち通信時間律速の条件を求めると，

$$\frac{n}{\sqrt{N}} < 421{,}766 \qquad (6)$$

である．ノード数：$N = 64$の場合，行列のサイズnは，

$$n < 3{,}374{,}128 \qquad (7)$$

となる．行列のサイズがこの条件を満たす場合は通信時間律速であり，本報告で提案したネットワークおよび通信アルゴリズムによる通信時間の短縮が実行時間の短縮に寄与する．また，上記の結果は 1 ノード当たりの演算能力$P =$ 1 TFlops，総帯域幅$B = 1.6$ Tbps $= 0.2$ TByte/s，すなわち，$B/P = 0.2$ Byte/Flopの場合であるが，仮にプラットフォームが変わってB/P 値が変化した場合，式(5)から上記の通信律速となる行列サイズはB/P 値に反比例して増減する．

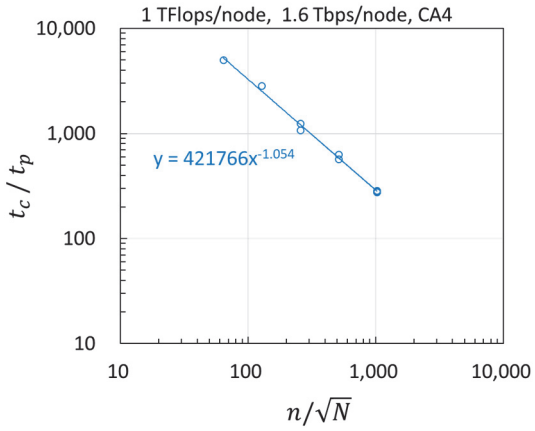

図 10　通信時間t_cと演算時間t_pの関係

5.2 2 次元光ハブによるスケーラビリティ

光ハブは，物理的には計算ノード間を波長ルーティングで接続している．そのため，接続できるノード数の上限は利用できる波長数によって制限され，約 80 程度である．これ以上のノード数を接続する方法としては，光ハブの多層化構成[4]，または多次元化構成がある．ここでは，並列行列積計算に適した 2 次元光ハブについて検討する．その構成を図 11 に示す．

計算ノードは仮想的に 2 次元グリッド状に配置され，各計算ノードはグリッドの同一行内のノード間通信を行う行ネットワークと，グリッドの同一列内のノード間通信を行う列ネットワークの 2 つのネットワークによって互いに接

続されている．これらの行および列ネットワークはそれぞれが光ハブであり，物理的には波長ルータを介したハブ接続，論理的にはフルメッシュ接続となっている．つまり論理トポロジーは，2 次元トーラス・ネットワークの各リング・ネットワークがフルメッシュ・ネットワークになった構成であり，2 次元の HyperX[15]と等価である．行ネットワークと列ネットワークは光学的には完全に分離されているため，波長の重複が許される．

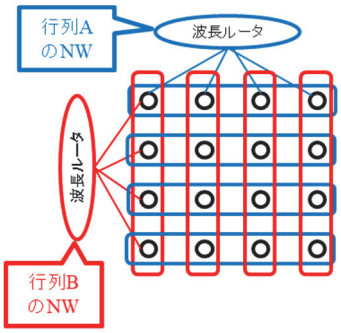

図 11　2 次元光ハブの構成

表 8 に 1 次元および 2 次元光ハブで 2 次元マッピングおよび CA4 アルゴリズムを用いた場合の SUMMA の通信時間理論値の比較を示す．ノード数Nおよび 1 ノード当たりの総帯域幅Bを固定して比較すると，2 次元光ハブでは 1 次元光ハブに比べて，1 ノードのポート数は$2/\sqrt{N}$倍減少し，1 ポート当たりの帯域幅bは$\sqrt{N}/2$倍増加し，必要な波長数は$1/\sqrt{N}$倍減少する．

一般に，2 次元光ハブで異なる行または列ネットワーク間の通信を行う場合は，2 次元トーラス等と同様にマルチホップが必要になるが，SUMMA 等の並列行列積計算では，小行列 $\mathbf{A}_{ij}$ の通信に行ネットワーク，小行列 $\mathbf{B}_{ij}$ の通信に列ネットワークを用いれば，ネットワーク間を跨ぐマルチホップは発生しない．また CA4 アルゴリズムの場合，1 次元光ハブでは小行列 $\mathbf{A}_{ij}$ の通信と小行列 $\mathbf{B}_{ij}$ の通信を同時に行うことは出来ないが，2 次元光ハブではそれぞれのネットワークが完全に分離しているため，同時に通信を行うことで通信回数を半分に減らすことが出来る．一方，2 次元光ハブでは中継ノードとして使えるノード数が 1 次元光ハブに比べて$1/\sqrt{N}$に減少するため，メッセージサイズ（データ長）は$\sqrt{N}$倍になる．

以上の効果を全て考慮すると，SUMMA の通信時間および実行時間が通信帯域律速の場合，表 8 の最下行に示す通り，1 次元光ハブおよび 2 次元光ハブを用いたときの SUMMA の実行時間は同じになると見込まれる．図 12 に SimGrid による SUMMA の実行時間のシミュレーション結果を示す．理論予測の通り，行列サイズが大きい場合に両者の実行時間はほぼ一致し，共に理論値に漸近していることが分かる．これは，波長数を固定で考えれば，2 次元光ハブを用いることで，通信時間を増やすことなく，接続できるノード数を波長数の 2 乗，すなわち 6400 程度までスケール出来ることを意味している．

表 8　1 次元光ハブおよび 2 次元光ハブによる SUMMA(2 次元マッピング，CA4)の通信時間の比較

ネットワーク	1 次元光ハブ	2 次元光ハブ
ノード数	N	N
1 ノードの総帯域幅	B	B
必要波長数	N	$\sqrt{N}$
1 ノードのポート数	N	$2\sqrt{N}$
1 ポートの帯域幅b	B/N	$B/(2\sqrt{N})$
通信回数M	$2\sqrt{N}$	$\sqrt{N}$
データ長s	n^2/N^2	$n^2/(N\sqrt{N})$
相対通信時間 $t_c = Ms/b$	$2n^2/(\sqrt{N}B)$	$2n^2/(\sqrt{N}B)$

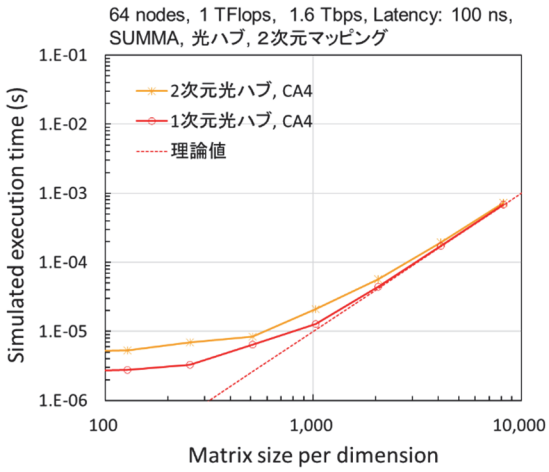

図 12　1 次元および 2 次元光ハブによる SUMMA 実行時間のシミュレーションの比較

5.3 コスト見積

1 次元および 2 次元光ハブを用いた並列計算システムのノード間ネットワークの概算のコスト見積を行った．コスト見積のためのそれぞれのネットワークの仕様を表 9 に示す．また，従来のネットワークとして，InfiniBand HDR 200 Gbps（以下，IB）のコストも見積った．IB は，主にラック間を想定してノードとスイッチ間の接続に Active Optical Cable(AOC)を用いたものと，主にラック内を想定して Direct Attached Copper(DAC)ケーブルを用いたものを見積もった．図 13 に，各ネットワークの帯域幅当たりのコストおよびその内訳を($/Gbps)単位で示す．各コストの算出方法については，付録 A.1 に示す．IB の場合，ラック内では Host Channel Adapter(HCA)の比率が大きく，ラック間では光トランシーバ（AOC）の比率が大きい．光ハブの場合，光トランシーバおよび FPGA のコストは 1 次元と 2 次元で共通であるが，2 次元化することで光源および光アンプの

コストをより多数のチャンネルでシェアすることになり，その部分のコスト低減が全体の低コスト化に寄与する．IB と光ハブを比較した場合，完全に同じ基準での比較とはなっていない点には注意が必要ではあるが，少なくとも光ハブのコストが IB に対して大幅に大きくなることは無いと見込んでいる．

表 9　コスト見積のためのノード間ネットワークの仕様

ネットワーク	1 次元光ハブ	2 次元光ハブ
1 ノードの帯域幅	800 Gbps	800 Gbps
1 ノードのポート数	32	32
1 ポートの帯域幅	25 Gbps	25 Gbps
ノード数	32	256
総帯域幅	25.6 Tbps	204.8 Tbps
波長数	32	16

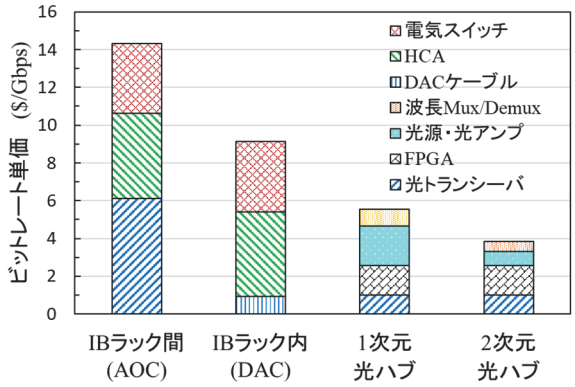

図 13　ノード間ネットワークの概算コスト見積

5.4 ネットワーク・キャッシュ

表 3 に示す通り，CA4 の相対性能（通信時間の逆数）は，CA1 に対してN倍になる．CA3 のように単にマルチパス・ルーティングを行うだけでは相対性能は$N/2$倍にしかならないが，図 7 に示す通り，1 度 Alltoall でNノードに分散配置したデータを繰り返し再利用することで更に 2 倍の高速化をしている．この効果は，メモリアクセスにおけるキャッシュの働きと似ているため，我々はネットワーク・キャッシュと呼んでいる．つまりフルメッシュ・ネットワークでは，ある 1 ノードに配置されたデータをダイレクト・ルーティングで別のノードに転送する場合に比べて，データをN分割してNノードにネットワーク・キャッシュし，必要に応じて目的のノードに Gather または Allgather することで，N倍高速に転送することが出来る．また，3.5 節で示した通り，CA4 通信アルゴリズムを用いることで，他の方式に比べて極めて高いメモリ容量当たりの相対性能を実現できる．同様のアルゴリズムを用いることで，並列行列積計算以外でも Send/Recv, Bcast, Reduce, Allreduce 等の通信を繰り返す計算では，同等の効果が期待できると考えている．

6. まとめ

これまでに，並列計算におけるノード間通信ボトルネックを解消するため，シリコンフォトニクスを用いた高帯域密度光トランシーバ「光 IO コア」，波長ルーティングを用いたノード間インターコネクト「光ハブ」とそれに適した「マルチパス・ルーティング」等を提案した．

並列行列積計算は応用範囲の広い計算であり，その実行時間がノード間通信の通信時間に律速され易い．そこで今回，光ハブによる並列行列積計算の高速化を検討した．

具体的には，並列行列積計算（SUMMA）に対して，マルチパス・ルーティングによるメッセージサイズの縮小と，フルメッシュ接続の全リンクを同時に利用することによる通信回数の削減が可能な光ハブに適した通信アルゴリズムを提案し，その通信時間短縮効果を理論およびシミュレーションで検証した．また，必要となるメモリ容量についても検討した．

その結果，ネットワークを光ハブに固定し，通信アルゴリズムの違いによる実行時間の比較を行った場合，理論とシミュレーションの結果は良く一致し，最も高速な通信アルゴリズム（CA4）では，標準 MPI を用いた場合に比べてノード数倍以上の高速化を確認した．また，通信アルゴリズムおよびネットワーク仕様（総帯域幅，リンク遅延時間等）を固定し，ネットワーク・トポロジーの違いによる比較を行うと，光ハブは 2D-torus に比べて約 7 倍，Fat-tree に比べて約 1.5 倍高速であることを確認した．また，上記の最も高速な通信アルゴリズムで必要とされるメモリ容量は，標準アルゴリズムの高々1.4 倍とそれほど大きくなく，必要とされるメモリ容量当たりの相対性能が突出して高いことを確認した．

上記の効果が得られるためには，通信時間が演算時間より長く，実行時間が通信律速である必要がある．シミュレーション結果から通信時間と演算時間を分離し，そのための条件を示した．また，計算ノード数を利用可能な波長数の 2 乗の 6400 程度までスケールさせることが可能な 2 次元光ハブの構成についても示した．また，ノード間ネットワークの概算コストを見積り，少なくとも従来の InfiniBand に比べて大幅なコストアップにならないこと，2 次元光ハブにすることにより，より低コスト化が期待できることを示した．最後に，フルメッシュ・ネットワークにおけるより一般化された通信高速化方法として，ネットワーク・キャッシュという概念を紹介した．今後，このネットワーク・キャッシュによる高速化を他のアプリケーションにも拡張して行きたいと考えている．

謝辞　この成果は，国立研究開発法人新エネルギー・産業技術総合開発機構（NEDO）の委託業務（JPNP13004）の結果得られたものです．日頃ご指導頂く大学共同利用機関

法人情報・システム研究機構アーキテクチャ科学研究系の鯉渕准教授に感謝致します.

参考文献

[1] Rumleya, S. et al.. Optical interconnects for extreme scale computing systems. Parallel Computing, 2017, vol.64, p.65-80.
[2] Nakamura, T. et al.. Fingertip-Size Optical Module, “Optical I/O Core”, and Its Application in FPGA. 2019, IEICE Trans. Electron., vol.E102–C, no.4, p.333-339.
[3] 賣野豊, 水谷健二, 臼杵達哉, 中村滋. ノード間波長ルーティング・インターコネクト（光ハブ）を用いた並列計算システム. 信学技報, 2019, CPSY2019-33, DC2019-33.
[4] Urino, Y. et al.. Wavelength-routing interconnect "Optical Hub" for parallel computing systems. 2020, HPC2020, p.81-91.
[5] Van de Geijn, R. A. et al.. SUMMA: Scalable universal matrix multiplication algorithm. Concurrency: Practice and Experience, 9(4), p.255-274, April 1997.
[6] Okamoto, K. et al.. 32×32 arrayed-waveguide grating multiplexer with uniform loss and cyclic frequency characteristics. Electron. Lett., 1997, vol.33, no.22, p.1865–1866.
[7] Noguchi, K. et al.. Field Trial of Full-Mesh WDM Network (AWG-STAR) in Metropolitan/Local Area. 2004, J. Lightwave Technology, vol.22, no.2, p.329-336.
[8] 水谷健二, 山口博史, 賣野豊. 光ハブによる Accelerator 間フルメッシュ接続に向けた Distributed Switch を介したメモリ間通信特性. 信学技報, 2019, CPSY2019-32, DC2019-32.
[9] 水谷健二, 山口博史, 賣野豊. 光ハブ用 Distributed Switch による複数ノード上 HBM2 メモリ間の集合通信特性. 2020, SWoPP2020 HPC 研究会にて発表予定.
[10] Cannon, L. E.. A cellular computer to implement the Kalman Filter Algorithm. PhD thesis, Montana State University, 1969.
[11] Fox, G. et al.. Solving Problems on Concurrent Processors, volume I. Prentice Hall, 1988.
[12] Solomonik, E. et al.. Communication-optimal parallel 2.5D matrix multiplication and LU factorization algorithms. Proc. Euro-Par2011, Lecture Notes in Computer Science, vol.6853, pp.90-109.
[13] http://simgrid.gforge.inria.fr/.
[14] https://jp.mellanox.com/products/infiniband-switches/QM8700.
[15] Ahn, J. H. et al.. HyperX: Topology, Routing, and Packaging of Efficient Large-Scale Networks. Proc. of the Conference on High Performance Computing Networking, Storage and Analysis, 2009, pp.1-11.
[16] https://store.mellanox.com/

付録

付録 A.1 ノード間ネットワークのコスト見積

1 次元および 2 次元光ハブのコストの内訳を表 A.1 および表 A.2 に示す．ただしこの表の単価は，我々が光ハブを構築する際に業者等から得た情報を基に，量産時の単価を予測したものであり，現時点でこの単価で調達できている訳ではない．ここで，AWG はアレイ導波路回折格子（Arrayed Waveguide Gratings）の略である[6][7]. また, FPGA のコストに関しては，全体のコストの 12.5%がネットワーク分，残りがアクセラレータ分であると見込んだ.

InfiniBand(IB) HDR の各部品の単価は Mellanox 社のストアサイト[16]の価格（2020 年 6 月 29 日時点）を用いた．電気スイッチに関しては, 1 次元光ハブの総帯域幅（25.6 Tbps）と総帯域幅が等しくなるように，40-port-HDR(200 Gbps)スイッチ(MQM8700HS2F)を 3.2 台用いるものとした. HCA のコストは，MCX654106A-HDAT-SP(Dual port)を 64 台，AOC または DAC ケーブルのコストは，それぞれ MSF1S00-H030E(30m)または MCP1650-H001E30(1m)を 128 本用いるものとして見積もった.

表 A.1　1 次元光ハブのコスト内訳

ボード	部品	単価（$）	数量	小計（$）
光源	レーザ	400	32	12,800
	光アンプ	500	32	16,000
	光スプリッタ	270	32	8,640
波長ルータ	AWG	180	64	11,520
計算ノード	光トランシーバ	25	1,024	25,600
	光アンプ	500	32	16,000
	AWG	180	64	11,520
	FPGA	1,250	32	40,000
合計（$）				142,080
総帯域幅（Gbps）				25,600
ビットレート単価（$/Gbps）				5.6

表 A.2　2 次元光ハブのコスト内訳

ボード	部品	単価（$）	数量	小計（$）
光源	レーザ	400	16	6,400
	光アンプ	500	16	8,000
	光スプリッタ	432	16	6,912
波長ルータ	AWG	90	1,024	92,160
計算ノード	光トランシーバ	25	8,192	204,800
	光アンプ	500	256	128,000
	AWG	90	512	46,080
	FPGA	1,250	256	320,000
合計（$）				789,312
総帯域幅（Gbps）				204,800
ビットレート単価（$/Gbps）				3.9