Vol.2009-HPC-121 No.38
2009/8/6

マルチコアクラスタ向け通信手法を用いた密度行列繰り込み群法の並列化

山田 進[†1,†3] 今村俊幸[†2,†3]
奥村雅彦[†1,†3] 町田昌彦[†1,†3]

量子多体系シミュレーションに用いられる密度行列繰り込み群法は，本来，1 次元モデル用に開発された計算手法であるが，著者らによる並列化により，2 次元モデルに適用することが可能になった．この並列化では多くの all-to-all 通信を必要とする．しかし，現在，並列計算機の主流であるマルチコアクラスタは，ネットワークの帯域に対してコア数が多い．そのため，全てのプロセスで通信を行う all-to-all 通信はマルチコアクラスタに適していない．そこで，著者らはデータの分割方法を再構成することで all-to-all 通信を用いない並列化手法を提案する．また，実際に T2K オープンスパコン（東大）を利用したシミュレーションから，提案手法の有効性を確認する．

Parallelization of Density Matrix Renormalization Group Method using Communication Strategy for Multicore Cluster

Susumu Yamada,[†1,†3] Toshiyuki Imamura,[†2,†3]
Masahiko Okumura[†1,†3] and Masahiko Machida[†1,†3]

The density matrix renormalization group (DMRG) method is widely used for solving large quantum lattice systems. While the DMRG method has been originally developed for a one-dimensional model, the DMRG method has been extended to two-dimensional models by our parallelization scheme. However, the parallelization scheme requires a lot of all-to-all communication. Such a communication is not suitable for a multicore cluster which is a mainstream type of parallel computer architecture. Therefore, we propose a new parallel algorithm which do not need the all-to-all communication operation. We confirm that the new algorithm actually achieves good performance on a multicore cluster.

1. はじめに

近年注目を集めている高温超伝導体や強磁性体などの固体材料が示す興味深い現象の多くは，電子間に強い相関が働く強相関量子多体効果によるものと考えられている．しかし，量子多体問題を解析的に解くことは困難であるため，数値シミュレーションが有力な手法として注目されている．多くの物理研究者たちはこの現象を理解するために，**図 1** のハバードモデル呼ばれる格子上に存在する電子間の短距離相互作用のみを考慮したモデルに注目している[1),2)]．このハバードモデルの数値シミュレーション手法の 1 つに，重要な成分だけを考慮してシミュレーションを行う密度行列繰り込み群法（DMRG 法）がある（**図 2** 参照）[3),4)]．この DMRG 法は短距離相互作用のみを考慮した 1 次元モデルのエネルギーを表現するハミルトニアン行列の基底状態（最小固有値とそれに対応した固有ベクトル）を高精度で計算する方法であり，実際 1 次元モデルに対しては大きな成功を収めてきた．

しかし，実際の固体材料では，その 2 次元的構造に興味深い物性発現の要因が潜んでおり，成果を得るには**図 3** のように格子を 2 次元に配置した 2 次元ハバードモデルをシミュレーションすることが必須である[5)]．そのため，DMRG 法を 2 次元モデル用に拡張する方法が提案されてきた．その代表的な方法として**図 4** のように 2 次元モデルをジグザクの 1 次元モデルとみなしてシミュレーションを行う multichain 法および**図 5** のように DMRG 法を直接 2 次元モデル用に拡張する direct extension 法がある．multichain 法は 1 次元モデルを計算するのとほぼ同じ計算量でシミュレーションできる．しかしながら，図 4 のように長距離相互作用を考慮する必要があるため，精度に関して問題があることが指摘されている[6),7)]．一方，direct extension 法は短距離相互作用を保ったまま 2 次元に拡張しているため，高精度のシミュレーションが期待できるが，2 次元方向へのモデルの拡大に伴って計算量が指数関数的に増加する．そのため，2 次元モデルのシミュレーションには並列化が必要であるが，著者らにより並列化手法が提案され実際に SGI Altix3700Bx2 を利用した並列計算において 100 プロセッサを越える並列計算おいても並列化の効果を得ることに成功している[8),9)]．

†1 日本原子力研究開発機構
Japan Atomic Energy Agency
†2 電気通信大学
The University of Electro-Communications
†3 独立行政法人科学技術振興機構
CREST(JST)

Vol.2009-HPC-121 No.38
2009/8/6

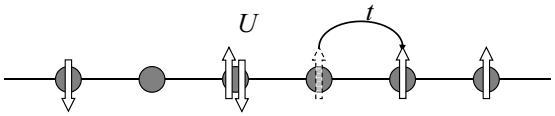

図 1 1 次元ハバードモデルの模式図．●は格子点，矢印はスピン状態の異なる電子 (フェルミ粒子) を表している．ハバードモデルではフェルミ粒子を対象としているため，1 つの格子点に同じ向きのスピンの粒子は 1 つしか存在できない．U および t はそれぞれ 1 つの格子点に異なる向きのスピンの粒子が存在する場合の斥力および粒子のホッピングパラメータである．

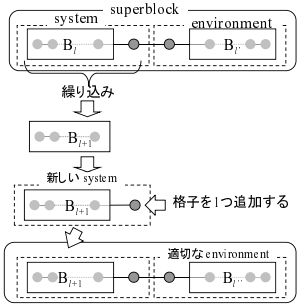

図 2 1 次元モデルに対する密度行列繰り込み群法の模式図．system と environment から構成されており，全体を superblock と呼ぶ．system の状態を繰り込むことで格子を 1 つ追加しても全体の状態数が増加しないようにしている．

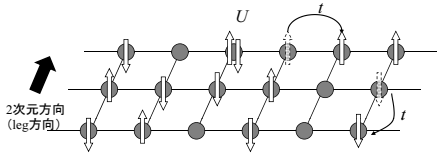

図 3 2 次元 (3-leg) ハバードモデルの模式図．2 次元方向の格子数を leg 数と呼ぶ．

しかし，この提案した並列化手法は全てのプロセス間で通信を行う all-to-all 通信を必要とするため，ネットワークの帯域に負担がかかる．特に，最近主流な並列計算機であるマルチコアクラスタは，以前の並列計算機と比較しネットワークの帯域に対するコア数が多く all-to-all 通信には適していないと考えられる．そのため，マルチコアクラスタの性能を有効に利用するには all-to-all 通信を用いないアルゴリズムを採用することが重要であると推測される．そこで本研究では，データの通信パターンに着目し，このパターンを適切に組み合わせることで全てのプロセス間の all-to-all 通信を必要としない通信アルゴリズムを提案する．また，実際にマルチコアクラスタを利用したテスト計算によりその有効性を確認する．

以下，第 2 章では DMRG 法およびその並列化に付いて説明し，第 3 章では全プロセス間の通信を行わないデータの分割方法および通信方法を提案する．第 4 章では，実際に並列計算機を利用したテスト計算でその有効性を示す．第 5 章はまとめである．

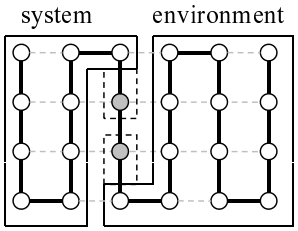

図 4 2 次元モデルに対する multichain 法の模式図．実線に沿って DMRG 法を適用することで，1 次元モデルとしてシミュレーションが実現できるが，1 次元モデルとみなすことで破線で表した格子間の相関は長距離になる．

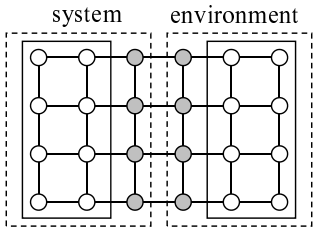

図 5 2 次元モデルに対する direct extension 法の模式図．直接 2 次元モデルに拡張しているため，全ての格子間の相関は短距離のままである．

2. 並列 DMRG 法のアルゴリズム

2.1 2 次元モデルに対する DMRG 法のアルゴリズム

ハバードモデルを計算するためには、モデルのエネルギーを表現するハミルトニアンの基底状態を計算すれば良い．しかし，このハミルトニアンの次元は電子の取りうる状態数に等しく，モデルの格子数や電子数の増加に伴って，指数関数的に増加する．この困難を回避するため，S.R.White は図 2 のように 1 次元モデル全体（superblock）を system と environment と分割し，system の重要な状態をのみを考慮することで，全体の状態数を一定に保ったまま system を大きくできる DMRG 法を提案した[3),4)]．DMRG 法の具体的な計算方法は以下の通りである．

(1) superblock に対応するハミルトニアン行列を構成し，その基底状態を計算する．
(2) （1）で求めた基底状態を利用し，system に対応した密度行列を構成する，
(3) 密度行列を対角化し，大きい方から m 個の固有値とそれに対応する固有ベクトルを計算する（m は考慮する状態数）．
(4) （3）で得られた固有ベクトル（状態）を利用して，system を重要な状態で近似する．さらに，この system に新しい格子点を追加する．
(5) 適切な environment を利用して新しい superblock を構成して（1）へ戻る．

この DMRG 法は図 2 のように本来は 1 次元モデル用に提案された方法である．より現実的なモデルである 2 次元モデルを扱うためには拡張する必要があり，代表的な方法に 2 次元モデルをジグザグの 1 次元モデルとみなして計算する multichain 法（図 4 参照），およ

び直接 2 次元モデルに拡張する direct extension 法（図 5 参照，以下 dex-DMRG 法と記す）がある．multichain 法は 1 次元モデル用の方法と計算量はほぼ同じであるが，本来の DMRG 法には含まれない長距離相関を考慮する必要があり精度や収束性に関して問題があることが指摘されている[6),7)]．一方，dex-DMRG 法は直接 2 次元モデル用に拡張しているため，2 次元方向の格子点の拡大（leg 数の増加）に伴って計算量は指数関数的に増加するが，短距離相関のみを考慮しているため精度や収束性は優れている．そのため，高精度のシミュレーション結果を得るためには dex-DMRG 法を並列化する必要がある．

2.2 dex-DMRG 法の並列化

2 次元モデルに対する dex-DMRG 法においてメモリの使用量および計算量の多い部分は密度行列の固有値計算と superblock のエネルギーを表現するハミルトニアンの基底状態の計算である．前者は密行列（正確には各ブロック行列が密行列である対角ブロック行列）の固有値計算であるため，ScaLAPACK 等を利用することで効率的な並列計算が可能である．一方，後者については，疎行列であるハミルトニアン行列の基底状態（最小固有値およびそれに対応する固有ベクトル）を計算すればよいので，反復法を用いるのが望ましい．今回の研究では基底状態を計算する反復法として LOBPCG 法[10),11)] を採用している．反復法において最も計算量が大きい演算は行列とベクトルの掛け算であるため，この掛け算を並列化することが重要である．疎行列とベクトルの掛け算の並列化は，疎行列を行方向に分割して並列計算するのが一般的である．しかし，疎行列内の非零要素の分布に偏りがある場合には，演算量および通信量に偏りができてしまう。特に並列度が大きい場合にはその効果が得られなくなる．そのため，超並列計算を実現するためには演算量，通信量を均等に分割する並列化手法を提案する必要がある．

そこで著者らは，**図 6** の左図のように 4 つのブロックで構成されているモデルを図 6 の右図のように 3 つに分割できること，および，**図 7** のようにベクトル v がそのブロックの状態に対して階層構造になっていることを考慮して行列 V および V_c と変換することで，ハミルトニアン行列 H とベクトル v の掛け算 Hv が H_lV, H_cV_c, VH_r^T の 3 つの行列積に分割できることを見出した．ここで H_l, H_c, H_r は図 6 の右図のそれぞれ左，中央，右の各ブロックの状態から導かれるハミルトニアン行列である．

このとき，行列 H_l, H_c, H_r は疎行列であり，V および V_c は密行列である．列方向および行方向に分割した V をそれぞれ V^c および V^r，列方向に分割した V_c を V_c^c とする．また，各プロセスで H_l, H_c, H_r の全ての非零要素の情報を格納していると仮定すると，H_lV^c, $H_cV_c^c$, $V^rH_r^T$ の各掛け算は通信を行うことなく並列計算が可能である．ただし，V^c, V^r,V_c^c

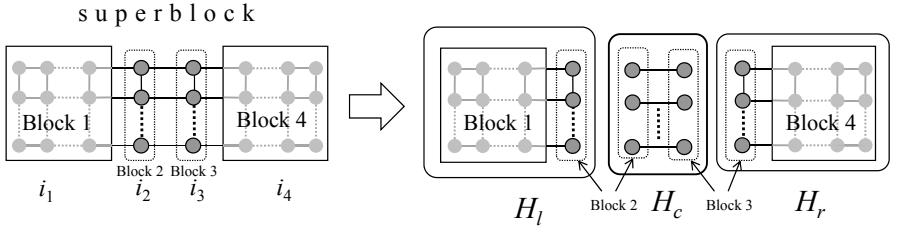

図 6 2 次元モデルの superblock の模式図．左図の 4 つのブロックで構成されている superblock を右図に示す 3 つのブロックに分解する．左図の各ブロックに対する状態を左から i_1，i_2，i_3，i_4 で表す．右図の 3 つのブロックに対応するハミルトニアン行列を左から H_l, H_c, H_r としている．

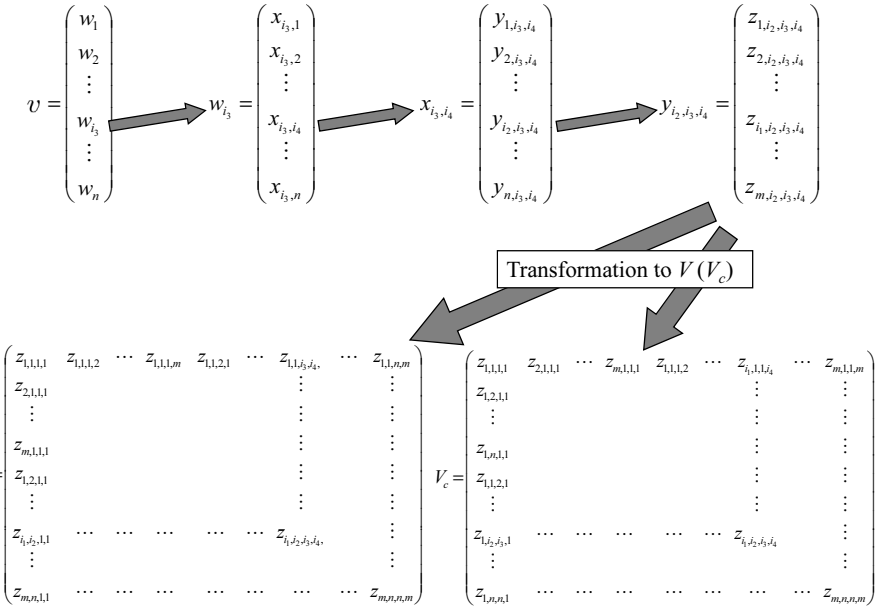

図 7 ベクトル v から行列 V および V_c への変換方法．ベクトル v は状態 i_1，i_2，, i_3 , i_4 に関して階層構造になっている．また，m は考慮する状態数（繰り込む状態数），n はハバードモデルでは $n = 4^s$ である．

間の変換を行うための通信が必要である．著者らはこの分割方法をによるハミルトニアン行列 H とベクトル v の積 Hv の並列計算アルゴリズムとして

CAL 1 $W_1^c = H_l V^c$
COM 1 V^c を V^r に変換するための通信
CAL 2 $Z_2^r = V^r H_r^T$
COM 2 Z_2^r を W_2^c に変換するための通信
COM 3 V^c を V_c^c に変換するための通信
CAL 3 $Z_3^c = H_c V_c^c$
COM 4 Z_3^c を W_3^c に変換するための通信
CAL 4 $W^c = W_1^c + W_2^c + W_3^c$

を考案した．ここで，上つきの添字 c と r はそれぞれ行列の分割方法が列方向，行方向であることを表現している．このとき，通信の COM1-4 は全て all-to-all 通信である．実際の計算では物理的に不適な状態に対するベクトル成分は無視するため，送信するデータ量はプロセスごとに異なる．そのため，上記の all-to-all 通信を実現するために，`MPI_alltoallv` を利用している．実際にこのアルゴリズムを用いて SGI Altix 3700Bx2 を利用したシミュレーションを実施したところ，100 プロセスを越える並列計算においても並列計算の効果を得ることに成功している[9]．

3. マルチコアクラスタ向け並列化

2.2 章で説明した通り，ハミルトニアン行列とベクトルの掛け算 Hv は 3 つの行列積 $H_l V$, $H_c V_c$, $V H_r^T$ に分解でき，行列 V および V_c を適切に分割することで並列化が実現できる．このとき，$H_l V$ および $V H_r^T$ を並列計算するためには行列 V をそれぞれ列方向および行方向に分割しており．この 2 つの分割した行列を直接変換するためには全プロセス間で通信を行う all-to-all 通信が必要である．

しかし，標準的な MPI の通信ライブラリである MPICH では，今回対象にしている DMRG 法のように通信量が多い場合には,「`isend-irecv` アルゴリズム」または「pairwise-exchange アルゴリズム」を利用する[12),13)]．これら 2 つの通信アルゴリズムは，どちらも通信量は最小であるが，全てのプロセスからそれ以外の全てのプロセスと直接通信するアルゴリズムである．最近主流のマルチコアクラスタのようにネットワークの帯域に比較しコア数が多い場合には，全てのプロセス間で通信を行うと，帯域を複数の通信で共有するため通信性能が低下する．また，総プロセッサ数 (コア数) が多くなると，通信回数が増加し通信のためのレイテンシの影響も大きくなる．そのため，ネットワークに高負荷がかからないような通信アルゴリズムを採用することが重要になる．そこで，全プロセス間での通信を必要としない通信アルゴリズムを提案する．

まず，行列 V の $((i_2-1)m+i_1, (i_3-1)m+i_4)$ 成分を $((i_1-1)n+i_2, (i_4-1)n+i_3)$ 成分に配置することで，新しい行列

$$Z = \begin{pmatrix} Z_{1,1} & Z_{1,2} & \cdots & Z_{1,m} \\ Z_{2,1} & & & \vdots \\ \vdots & & & \vdots \\ Z_{m,1} & \cdots & \cdots & Z_{m,m} \end{pmatrix}. \tag{1}$$

を構成する．このとき，m は考慮する状態数（繰り込む状態数）であり，n はハバードモデルでは $n = 4^s$（s は 2 次元方向の格子数）である．ここで Z_{i_1,i_4} は

$$Z_{i_1,i_4} = \begin{pmatrix} z_{i_1,1,1,i_4} & z_{i_1,1,2,i_4} & \cdots & z_{i_1,1,n,i_4} \\ z_{i_1,2,1,i_4} & & & \vdots \\ \vdots & & & \vdots \\ z_{i_1,n,1,i_4} & \cdots & \cdots & z_{i_1,n,n,i_4} \end{pmatrix}, \tag{2}$$

である．この行列 Z は行列 V の第 $(i_2-1)m+i_1$ 行を第 $(i_1-1)n+i_2$ 行に，第 $(i_4-1)n+i_3$ 列を第 $(i_3-1)m+i_4$ 列にそれぞれ置換した行列であるため，行列 Z の成分を列方向および行方向にそれぞれ分割して得られる行列 Z^c および Z^r の成分は，それぞれ H_l および H_r との掛け算を通信を行わずに実行できるが，この方法でも行列の分割を Z^c から Z^r に直接変換するためには all-to-all 通信が必要である．

ここで，Z_{i_1,i_4} の成分と行列 H_c の掛け算には通信が発生しない．そのため，**図 8** のように行列 Z を Z_{i_1,i_4} の大きさである n の倍数の大きさで 2 次元に分割した分割行列 Z^b の成分と行列 H_c の掛け算は通信を行わずに実現できる．つまり，Z^c, Z^r, Z^r の 3 つの分割行列に変換する通信を行うことで，ハミルトニアンとベクトルの掛け算 Hv を並列計算できる．しかし，これまでに述べた通り，Z^c から Z^r への直接の変換には all-to-all 通信が必要になる．そこで，**図 9** のような Z^b を経由する通信方法を採用する．このとき Z^c から Z^b への変換は図 9 で Z^b の縦一列のブロックを担当しているプロセス間のみで通信を行えばよい．同様に，Z^b から Z^r への変換は横一列を担当しているプロセス間のみで通信を行えばよい．つまり，$Z^c \to Z^b \to Z^r$ と変換することで，全プロセス間の通信を行う必要がない

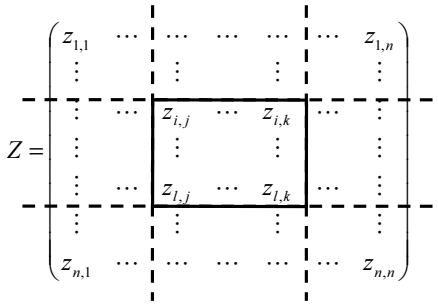

図 8 行列 Z を 2 次元分割する模式図．Z_{i_1,i_4} 単位で分割するように分割している．

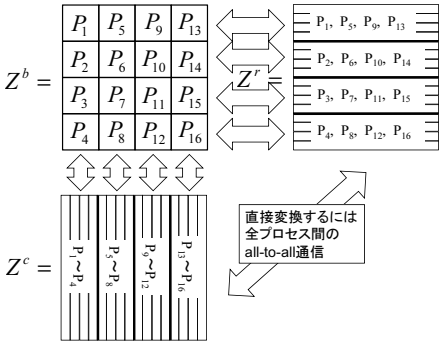

図 9 2 段階通信による Z^c, Z^b, Z^r の変換 (16 プロセスでの例)．P_1〜P_{16} は担当するプロセス番号を表している．直接 Z^c, Z^r 間で変換すると，all-to-all 通信が必要になる．一方，Z^c, Z^b 間の変換はプロセス 1〜4, 5〜8, 9〜12, 13〜16 間の通信で実現できる．また，Z^b, Z^r 間の変換はプロセス {1,5,9,13}, {2,6,10,14}, {3,7,11,15}, {4,8,12,16} 間の通信で実現できる．

（逆向きの変換も同様である）．以下，この方法を 2 段階通信（2-step communication）と呼ぶ．この通信方法において，図 9 の様にプロセスを割り当てて Z^c から Z^b の変換のための通信を行う際には連続した番号のプロセスを利用する．通常，連続したプロセス番号に対応するプロセッサ（コア）は物理的に近い位置にあるため高速な通信が期待できる．

この 2 段階通信アルゴリズム並列計算方法は

CAL 1 $W_1^c = H_l Z^c$
COM 1 Z^c を Z^b に変換するための通信
COM 2 Z^b を Z^r に変換するための通信
CAL 2 $Z_2^r = Z^r H_r^T$
COM 3 Z_2^c を W_2^b に変換するための通信
CAL 3 $Z_3^b = W_2^b + H_c Z^b$
COM 4 Z_3^b を W_3^c に変換するための通信
CAL 4 $Z^c = W_1^c + W_3^c$

とすることで実現できる[⋆1]．ただし，2.2 章でのベクトル v から行列 V への変換方法と，今回の Z への変換方法は異なっているため，2.2 章のハミルトニアンをそのまま利用するためには，Z^c では第 $(i_1-1)m+i_2$ 行の成分を第 $(i_2-1)n+i_1$ 行の成分に，Z^r では第 $(i_4-1)m+i_3$ 行の成分を第 $(i_3-1)n+i_4$ 行の成分に，Z^b では Z_{i_1,i_4} の (i_2,i_3) 成分を $((i_3-1)m+i_2,(i_4-1)n+i_1)$ 成分に置換する必要がある．この通信方法を用いたアルゴリズムでも各プロセスの総通信量は 2.2 章の方法とほぼ同じであることに注意が必要である．

4. 数 値 実 験

提案した通信方法の有効性を確認するため，シングルコア並列計算機である日本原子力研究開発機構所有の SGI Altix 3700Bx2 およびマルチコアクラスタである東京大学情報システムセンター所有の T2K Open Supercomputer[14] 上で並列計算を行う．2 つの計算機のスペック等を**表 1** に示す．本実験における計算対象のモデルは 3×10-格子ハバードモデル（アップスピン 14 個，ダウンスピン 14 個）であり，計算時間を**表 2** に示す．この結果から，これまでの通信方法（all-to-all 通信）を用いた場合，Altix では 256 プロセスまで高速化していることが確認できるが，T2K では 128 プロセスまでしか高速化しない．一方，提案した 2 段階通信を利用した 256 プロセスでの並列計算において，Altix では約 1.1 倍，T2K では約 1.6 倍の高速化を実現している．この結果から T2K においても 256 プロセスまで並列化の効果が得られることが確認できる．

ここで通信時間の詳細を確認するため，**表 3** に行列ベクトル積部分の通信時間を示す．今回のような行列データの転置のための all-to-all 通信では，プロセス数が増加するとプロセスあたりの通信量は減少するため，クロスバーネットワークでレイテンシが無視できるような理想的な計算機環境であるなら通信時間は減少する．しかし，Altix ではプロセス数を増加させでも all-to-all 通信の時間はほとんど変化しない．これは Altix ではプロセスあたりのネットワークの帯域がプロセス数の増加に伴って減少するためであると考えられる．一方，T2K はノード間はフルバイセクションでつながっているためプロセス数が増加しても，通信性能は低下しないため，プロセス数が 64 から 128 に増加すると通信時間は減少している．しかし，128 から 256 に増加すると通信時間が増加している．これは，T2K の通信のレイテンシが大きいためであると考えられる．

⋆1 このあとの数値計算において，COM1-4 の通信は `MPI_COMM_SPLIT` を利用してプロセスを分割していくつかのグループを作成し，そのグループ内で all-to-all 通信をすることで実現していることに注意が必要である．以下では，特に断らない限り all-to-all 通信は全てのプロセス間で行うものを指すこととする．

表 1 SGI Altix 3700Bx2 および T2K Open Supercomputer (Todai Combined Cluster) のスペック

	Altix	T2K
Processor	Intel Itanium2 (1.6GHz)	AMD Quad core Opteron 8356 (2.3GHz)
Number of processors per each node	128	4 (16 cores)
Network	NUMAlink	Myrinet-10G link
Compiler Compile option	Intel Fortran Compiler 11.0 `-O3 -fno-alias -ipo`	Hitachi Optimizing Fortran Compiler `-Oss -noparallel`

表 2 これまでの通信方法 (all-to-all) と提案した 2 段階通信 (2-step) の計算時間の比較

(a) SGI Altix 3700Bx2

Number of processors	Comm. time (sec)		Speedup
	all-to-all	2-step	
64	446.47	422.31	1.06
128	344.59	319.10	1.08
256	317.86	279.90	1.14

(b) T2K Open Supercomputer

Number of cores	Comm. time (sec)		Speedup
	all-to-all	2-step	
64	678.34	527.58	1.29
128	494.26	375.82	1.32
256	496.99	303.68	1.64

表 3 これまでの通信方法 (all-to-all) と提案した 2 段階通信 (2-step) の通信時間の比較

(a) SGI Altix 3700Bx2

Number of processors	Comm. time (sec)		Speedup
	all-to-all	2-step	
64	45.75	32.35	1.41
128	47.19	26.37	1.79
256	46.47	16.61	2.87

(b) T2K Open Supercomputer

Number of cores	Comm. time (sec)		Speedup
	all-to-all	2-step	
64	186.78	61.34	3.04
128	140.50	51.59	2.72
256	185.58	41.87	4.43

一方，2 段階通信を用いた場合，Altix および T2K のどちらの計算機においてもプロセス数の増加に伴って通信時間が減少していることが確認できる．また，all-to-all 通信を用いたケースと比較し少ない通信時間になっていることが確認できる．特に T2K の 256 プロセスの際には通信時間は約 4 分の 1 まで減少していることが確認できる．以上の結果から，今回提案した 2 段階通信はシングルコアおよびマルチコアどちらの並列計算機においても有効であるがマルチコアに対して特に有効であると結論付けられる．

5. ま と め

本論文では，all-to-all 通信を行わないない DMRG 法の並列化手法について報告した．これまでの DMRG 法の並列化では反復 1 回あたり 4 回の all-to-all 通信を行う必要があるが，all-to-all 通信は通信回数が多く，また，多くのプロセスでネットワークの帯域を共有するため，現在の並列計算機の主流であるマルチコアクラスタでは高性能計算が期待できない．実際，T2K を利用した並列計算において 128 コアまでしか並列化の効果が得られなかった．一方，提案した並列化手法は全プロセス間での通信がなく，また，通信回数も少ないためマルチコアクラスタにおいても高性能な通信を実現し，T2K において 256 コアまで並列化の効果が得られた．この高速化の効果はシングルコア計算機である Altix でも確認できた．この結果から，今回提案した通信手法はマルチコアクラスタ等のネットワーク帯域に対してプロセッサ（コア）数が多い計算機に有効であることがわかった．

1 つのプロセッサあたりのネットワーク帯域は十分大きく通信のレイテンシが小さい並列計算機では，並列化の際には通信量と演算量を均等に分割し，通信量を少なくすることで並列化性能を向上させることができた．しかし，マルチコアクラスタで並列化性能を向上させるためは，多少通信量が増えても，ネットワークの性能を考慮し，通信回数の少ない通信方法やネットワーク帯域の共有を少なくする通信方法を採用することが有効であると考えられる．

謝辞 本研究の一部は文部科学省科学研究補助金 (基盤研究 (C) 一般 20500044) の成果によるものである．

参 考 文 献

1) M. Rasetti, ed.: *The Hubbard Model: Recent Results*, World Scientific, Singapore (1991).
2) A. Montorsi, ed.: *The Hubbard Model*, World Scientific, Singapore (1992).
3) S. R. White: Density Matrix Formulation for Quantum Renormalization Groups, *Phys. Rev. Lett.* 69, pp.2863-2866 (1992).
4) S. R. White: Density-matrix algorithms for quantum renormalization groups, *Phys. Rev. B* 48, pp.10345-10355 (1993).
5) 今田正俊: 酸化物高温超伝導体はどこまでわかったか 理論の立場から，パリティ，丸善株式会社，Vol. 23, No. 4, pp.28-32 (2008).
6) R. M. Noack and S. R. Manmana: Diagonalization- and Numerical Renormalization-Group-Based Methods for Interacting Quantum Systems, *Proc. of AIP Conf.*, 789, pp. 91-163 (2005).
7) G. Hager, G. Wellein, E. Jackemann, and, H. Fehske: Stripe formation in dropped Hubbard ladders, *Phys. Rev. B*, 71, 075108 (2005).
8) S. Yamada, M. Okumura, and M. Machida: Direct Extension of Density-Matrix Renormalization Group toward 2-Dimensional Quantum Lattice Systems: Study for Parallel Algorithm, Accuracy and Performance, e-print arXiv:cond-mat/070.0159.

Vol.2009-HPC-121 No.38
2009/8/6

9) S. Yamada, M. Okumura, and M. Machida: High Performance Computing for Eigenvalue Solver in Density-Matrix Renormalization Group Method: Parallelization of the Hamiltonian Matrix-Vector Multiplication, J.M.L.M Palma et al. (Eds.):VECPAR 2008, LNCS 5336, pp.39-45 (2008).
10) A. V. Knyazev: Preconditioned eigensolvers - An oxymoron?, *Electronic Transactions on Numerical analysis*, Vol. 7, pp. 104-123 (1998).
11) A. V. Knyazev: Toward the optimal eigensolver: Locally optimal block preconditioned conjugate gradient method, *SIAM J. Sci. Comput.*, 23, pp.517-541 (2001).
12) MPI homepage: http://www.mcs.anl.gov/research/ projects/mpi/index.htm.
13) R. Thakur, R. Rabenseifner, W. Gropp: Optimization of Collective Communication Operations in MPICH, International Journal of High Performance Applications, Vol. 19, No. 1, pp.49-66 (2005).
14) T2K Open Supercomputer Alliance: http://www.open-supercomputer.org.