Vol.2014-HPC-145 No.17
2014/7/29

超高並列環境での通信削減を目的とした Chebyshev 基底共役勾配法の特性評価

熊谷洋佑[†1]　藤井昭宏[†1]　田中輝雄[†1]　須田礼仁[†2]

コア数の増大によりスーパーコンピュータの性能は著しく向上している．大規模な連立 1 次方程式を解く反復解法である共役勾配法（CG 法）は高並列な環境では演算処理にかかる時間は減少する．それに対して，集団通信による時間が増大し，問題となっている．近年，集団通信の回数を削減する手法が提案され，その中でも安定して CG 法と同等の反復回数で収束する Chebyshev 基底共役勾配法（CBCG 法）がある．本稿では疎行列ベクトル積や通信の隠蔽などの最適化を施した CBCG 法を超高並列環境で実装し，CG 法との比較を行い通信削減の特性を示す．

1. はじめに

大規模な科学技術計算の処理を行うのにスーパーコンピュータが用いられる．近年のスーパーコンピュータはコア数の増大とともに性能は向上している．並列に処理を行う際にノード間での通信が発生する．特にリダクションやブロードキャストなどの，全ノードがデータを送受信する集団通信はノード数が増えるほど通信に要する時間も増える．通信による遅延は物理的制約により一定以下にできないことからアルゴリズムの観点からの通信の削減が必要である．

一方，正定値対称な疎行列を係数にもつ連立一次方程式を解くのに反復解法である共役勾配法 (Conjugate Gradient method : CG 法) [1] が広く用いられる．CG 法では 1 反復中に 1 回の疎行列ベクトル積と 2 回の内積計算が発生する．この内積計算では集団通信が発生し，特に高並列環境においてボトルネックとなる．

そこで，内積計算の回数を削減した CG 法が提案されてきた．Chronopoulos らにより s-step CG 法[2]が提案され，Toledo の博士論文より Krylov Basis CG 法[3]が提案され，本谷らにより k 段飛ばし CG 法[4]が提案されている．しかし，これらの手法はアルゴリズム的には CG 法と同じ解法であるが，丸め誤差の影響を受け反復回数の増大または発散する欠点があった．Hoemmen や須田らにより Chebyshev 多項式を基底とし，Krylov 部分空間をまとめて生成する Chebyshev 基底共役勾配法（Chebyshev Basis Conjugate Gradient method : CBCG 法）を提案され，CG 法と同等の反復回数で収束することが報告されている[5][6][7].

本論文では，疎行列ベクトル積（SpMV）などの最適化や通信の隠蔽も施し，通信削減アルゴリズムである CBCG 法を高並列な環境での特性評価を行った．

以下，2 章では CG 法と CBCG 法のアルゴリズムについて，3 章で CG 法と CBCG 法の演算量と通信回数について，4 章で，今回実装した最適化手法について述べる．5 章では数値実験とその評価を行い，6 章ではまとめと今後の課題を述べる．

2. 共役勾配法と Chebyshev 基底共役勾配法

2.1 共役勾配法

CG 法の実装については参考文献[1]に基づいている．反復部のアルゴリズムを図 1 に示す.

```
1  repeat
2      if n = 0
3          p_n ← r_n
4      else
5          β_n ← (r_n, r_n) / (r_{n-1}, r_{n-1})
6          p_n ← r_n + β_n p_{n-1}
7      end if
8      α_n ← (r_n, r_n) / (p_n, A p_n)
9      x_{n+1} ← x_n + α_n p_n
10     r_{n+1} ← r_n − α_n A p_n
11     n ← n + 1
12 until ‖ r_n ‖/‖ r_0 ‖ < ε
```

図 1　CG 法のアルゴリズム

2.2 Chebyshev 基底共役勾配法

CBCG 法の実装については参考文献[7]に基づいている．CBCG 法の反復部のアルゴリズムを図 2 に示す．CBCG 法におけるkとは Krylov 部分空間をまとめて生成する本数としている．つまり，CBCG 法 1 反復は CG 法のk反復に相当することになる．

†1 工学院大学
Kogakuin University
†2 東京大学
The University of Tokyo

1	repeat
2	$S_{n+1} \leftarrow (T_0(A)\boldsymbol{r}_n, T_1(A)\boldsymbol{r}_n, \cdots, T_{k-1}(A)\boldsymbol{r}_n)$
3	if $n = 0$
4	$Q_{n+1} \leftarrow S_{n+1}$
5	Else
6	$B_{n+1} \leftarrow (Q_n^T A Q_n)^{-1} Q_n^T A S_{n+1}$
7	$Q_{n+1} \leftarrow S_{n+1} - Q_n B_{n+1}$
8	end if
9	$\boldsymbol{a}_{n+1} \leftarrow (Q_{n+1}^T A Q_{n+1})^{-1} Q_{n+1}^T \boldsymbol{r}_{nk}$
10	$\boldsymbol{x}_{(n+1)k} \leftarrow \boldsymbol{x}_{nk} + Q_{n+1}\boldsymbol{a}_{n+1}$
11	$\boldsymbol{r}_{(n+1)k} \leftarrow \boldsymbol{r}_{nk} - AQ_{n+1}\boldsymbol{a}_{n+1}$
12	$n \leftarrow n + 1$
13	until $\| \boldsymbol{r}_n \| / \| \boldsymbol{r}_0 \| < \varepsilon$

図 2　CBCG 法のアルゴリズム

3. 演算量と通信回数

この章では，1 節に CG 法と CBCG 法の演算量について，2 章で載せたアルゴリズムをもとに算出した結果を述べ，2 節に通信の発生回数について述べる．3 節に今回の CG 法と CBCG 法の予備実験として，SpMV と内積（Dot）のサイズを一定とし，コア数を増加させたときの時間の変化を示し，考察を述べる．

3.1 演算量

CG 法の主な計算は SpMV が 1 回と Dot が 2 回となる．

CBCG 法では Kryolov 部分空間を$(k-1)$次の Chebyshev 多項式で生成するため SpMV が$(k-1)$回．また，図 2 の 6 行目のQ^TASと 9 行目のQ^TAQにおいて疎行列 A とk本のベクトル S と Q の積がそれぞれ行われる．これは，SpMV をk回行うことと同じ演算量であるため，CBCG 法は SpMV を$(3k-1)$回行うことになる．さらにQ^Tとの積が行われる．これは，　k^2回の Dot と同じ計算となる．最後に相対残差の計算をするために Dot が 1 回行われる．つまり，Dot を(k^2+1)回行うことになる．他にもベクトル演算やk次の方程式の直接解法などの細かい計算はあるが，代表的な SpMV と Dot のみで考える．CBCG 法の 1 反復は CG 法のk反復に相当するため，CBCG 法の計算回数に対してそれぞれkで割った CG 法 1 反復あたりの計算回数を表 1 に示す．SpMV だけでも CBCG 法の演算量は約 3 倍になることがわかる．

表 1　CG 法 1 反復あたりの計算回数

	CG	CBCG
SpMV	1	$3 - \frac{1}{k}$
Dot	2	$k + \frac{1}{k}$

3.2 各手法の通信回数

CG 法の反復中において通信が発生する部分は SpMV による隣接通信 1 回と，Dot による集団通信が 2 回である．

CBCG 法では，図 2 の 2 行目において$(k-1)$次の Chebyshev 多項式で SpMV が$(k-1)$回. また, 6 行目のASと 9 行目のAQの計算で隣接通信が発生するため，合計で$(k+1)$回の隣接通信が発生する．集団通信は相対残差を計算するために Dot が 1 回と，Q^TAQとQ^TASでそれぞれサイズkの行列のリダクションを行う必要があるため，合計で 3 回となる．CBCG 法の通信回数を CG 法の回数に換算した 1 反復あたりの通信回数を表 2 に示す．CBCG 法は隣接通信の回数は多くなるが，集団通信に関してはkを増やすほど回数が減ることになる．

表 2　CG 法 1 反復あたりの通信回数

	CG	CBCG
隣接通信	1	$1 + \frac{1}{k}$
集団通信	2	$\frac{3}{k}$

3.3 予備実験　SpMV と内積の実行時間

ここで，100^3 の立方体領域における三次元拡散方程式を使い SpMV と Dot のコア数を増加させた時の実行時間の変化をそれぞれ図 3 に示す．実験環境は 5.1 節に記す．Dot は実行時間が減少することなく．停滞していることがわかる. SpMV に関しては 1536 コアまでは減少傾向にあったものが，高並列になるに従い実行時間が停滞する結果となっている．

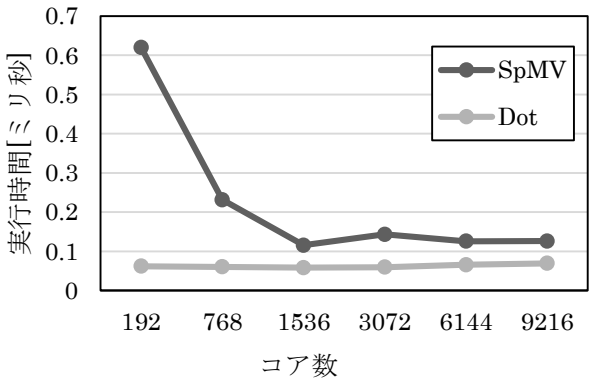

図 3　コア数の増加による SpMV と Dot の時間の変化
（サイズ 100^3）

9216 コアで実行したときの SpMV と Dot について演算と通信の割合を図 4 に示す.

Dot に関しては実行時間のほとんどが通信時間によって占められている．SpMV は計算時間と通信時間が半分ずつ

占めている．つまり，高並列な環境では計算量ではなく通信時間によって実行時間が左右されると考えられる．

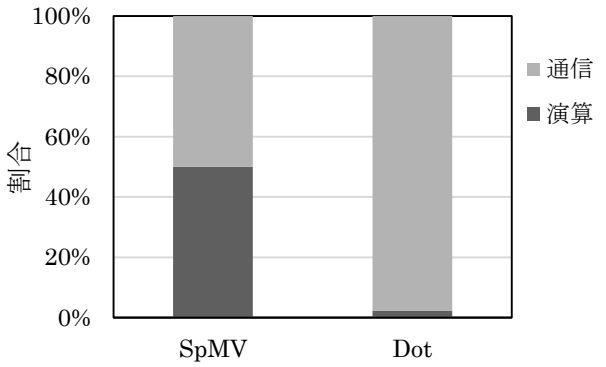

図 4　9216 コアにおける SpMV と Dot の時間の割合

4. 最適化手法

4.1 対称行列向け疎行列ベクトル積

CG 法は正定値対称な行列を係数にもつ連立 1 次方程式を解く反復解法である．そのため，問題行列Aは対角行列をD，下三角行列をLとすると，

$$A = LDL^T$$

と表すことができる．

一方，問題規模が大きくなるほど行列Aへの参照の負荷も大きくなる．行列の対称性を利用することで，行列を対角部分と下三角部分のみ保持し，SpMV を高速化する．

疎行列ベクトル積を行う際に，下三角行列を計算すると同時に対応する上三角部分も計算を行うことで行列Aの各要素の参照回数を約半分にすることができる．

4.2 疎行列と複数本のベクトルの積

CBCG 法において疎行列と複数本のベクトルの積（SpMM）が現れる．これは，ベクトルの本数がk本だとすれば SpMV をk回行うことになる．単純に SpMV を行えば問題行列Aへの参照もk回発生する．

そこで，問題行列Aへの参照を 1 回で複数本のベクトルとの積をまとめて行うことで，高速化を施す．また，隣接通信も一度にまとめて行っている．

5. 数値実験

この章では，1 節に今回使用した実験環境について，2 節に予備実験として今回使用する問題で CBCG 法が安定して収束することを示す．3 節において問題規模を一定としてコア数を増加させたときの CG 法と CBCG 法の実行時間の推移を示し，考察する．

5.1 実験環境

本実験では，FX10 スーパーコンピュータシステム（東京大学）[8] を使用し，数値実験を行った．FX10 の構成を表 3 に示す．数値実験は，最大 768 ノード，12288 コアを使用し，1 コアに 1 プロセス立ち上げる FlatMPI で行った．

問題には立方体領域における三次元拡散方程式となる拡散係数が一定の等方性問題，Z 軸方向の拡散係数が他軸の 100 倍となる異方性問題，ダルシー流れ問題から導出される拡散係数が不連続に変化する問題（ダルシーの流体問題）の 3 種を使用した．問題行列の分散には ParMETIS[9]を利用した．問題には対角スケーリングを施し，実験を行った．CBCG 法で必要となる最大固有値をべき乗法で求めた値，最小固有値を 0 に固定した．反復の終了条件は 2 ノルム相対残差が 10^{-12} 未満になったときとした．

表 3　FX10 の構成

ノード	CPU 数	1
	メモリ	32GB
CPU	SPARK64^{TM}IXfx	
	コア数	16 Core / CPU
	動作周波数	1.848GHz
コンパイラ	富士通社製 C コンパイラ	

5.2 実験 1　CBCG 法の収束性

今回使用する問題 3 種において CBCG 法が CG 法と同等の反復回数で収束することを示す．各問題での反復回数の比較をそれぞれ図 5，図 6，図 7 に示す．問題サイズはそれぞれ 100^3 とし，CBCG 法のkは 1 から 50 までとした．また，CBCG 法の反復回数は CG 法に相当する反復回数に換算している．

どの問題においても CG 法と CBCG 法の反復回数がほとんど同じ結果となった．等方性問題においてはkを 40 近くにすると反復回数が少し増えているが，発散することはなかった．これより，CBCG 法は安定して CG 法と同等の反復回数で収束することがわかる．

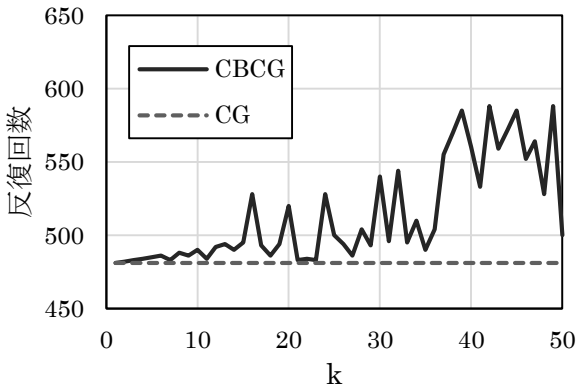

図 5　等方性問題における反復回数

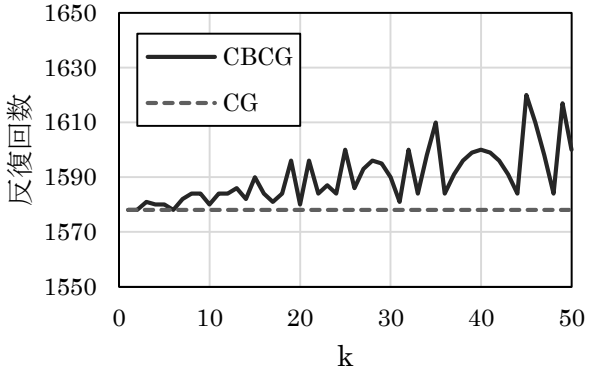

図 6　異方性問題における反復回数

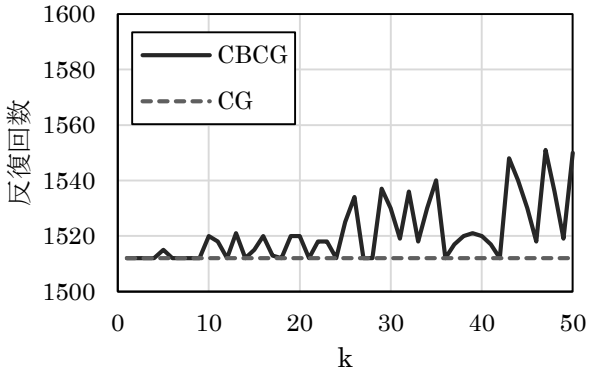

図 7　ダルシーの流体問題における反復回数

5.3 実験 2　ストロングスケーリングでの実験結果

実験 1 の結果より 3 種の問題で CBCG 法の反復回数が CG 法と同等であった. 3 種の問題は行列の構造が同じであるため，実行時間の計測にはダルシーの流体問題のみを用いた．比較手法として CG 法と CBCG 法の$k = 10, 15, 20$の 4 つの手法で実験を行った．今回の計測実験では，問題サイズを一定とし，コア数を増加させたストロングスケーリングでの計測を行った．最初に問題サイズ 100^3 での結果を示し，次に 200^3 での結果を示す．最後に考察を述べる．

5.3.1 問題サイズ 100^3 での実験結果

問題サイズ 100^3 での反復回数を表 4 に，コア数を増加させたときの実行時間の推移を図 8 に示す.

768 コアでは CBCG 法の実行時間が CG 法の約 3 倍となっているため，通信時間よりも計算時間が大半を占めていることが考えられる．また，CBCG 法の演算量が CG 法の約 3 倍であるため，通信による影響が少ないとすれば妥当であるといえる. 3072 コア以降は両者ともにほとんど同じ実行時間となっている．CG 法に関しては実行時間が停滞している状態となっている．

表 4　ダルシーの流体問題 100^3 における反復回数

	反復回数
CG	1512
CBCG(10)	1520
CBCG(15)	1515
CBCG(20)	1520

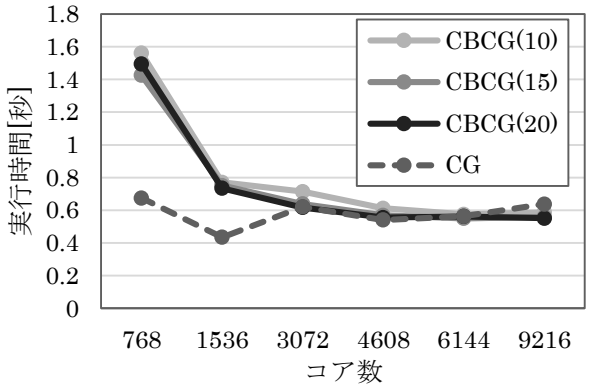

図 8　コア数増加による実行時間の推移（問題サイズ 100^3）

1536 コアと 9216 コアのときの各手法の演算と通信の内訳を図 9，図 10 にそれぞれ示す．時間計測は全プロセスが計測した時間の平均値から出している．

CG 法では通信が全体の半分以上を占めている．1536 コアのとき，CBCG 法は演算時間が約 0.5 秒に対し，CG 法の実行時間は約 0.4 秒で，演算時間だけでも CBCG 法は CG 法以上の時間となっている．9216 コアのときには，CBCG 法の演算時間が削減され，全体の実行時間で上回る結果となった．

また，CG 法の計算時間は 1536 コアと 9216 コアではあまり変化していないことがわかる．しかし，通信時間が増えたことにより，全体の実行時間も増えているため，CG 法は 1536 コア以上に並列化しても時間が増大し続けることになる．

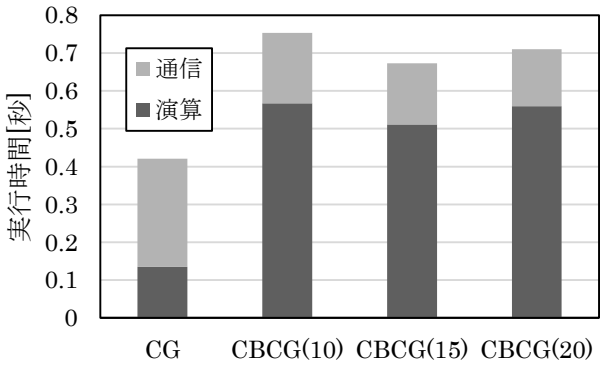

図 9　1536 コアでの実行時間の内訳（問題サイズ 100^3）

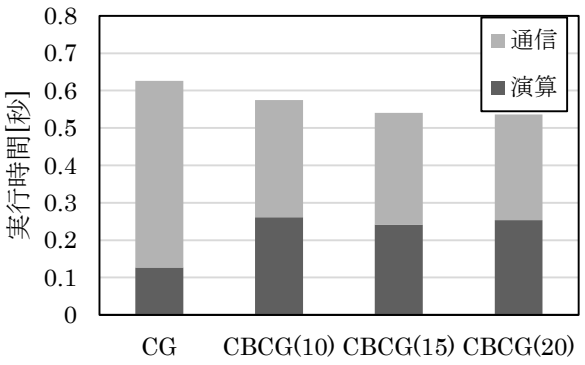

図 10　9216 コアでの実行時間の内訳(問題サイズ 100^3)

1536 コアと 9216 コアの通信時間の比較を図 11 に示す．どの手法においても 1536 コアのときよりも 9216 コアの通信時間が約 1.7~1.8 倍増加している結果となった．9216 コアのときの CG 法と CBCG 法を比べると CG 法が CBCG 法の約 1.5~1.7 倍の通信時間をとなった．

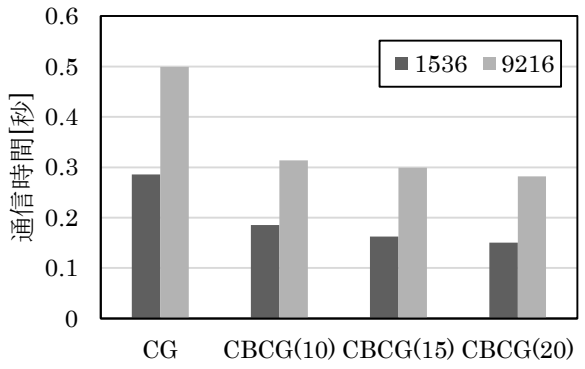

図 11　1536 コアと 9216 コアの通信時間(問題サイズ 100^3)

5.3.2 問題サイズ 200^3 での実験結果

問題サイズ 200^3 での反復回数を表 5 に，コア数を増加させたときの実行時間の推移を図 12 に示す．

3072 コアまでは CBCG 法の実行時間が CG 法の約 3 倍かかっている．12288 コアでは CG 法と CBCG 法がほぼ同じ実行時間となっている．また，CG 法は 12288 コアから実行時間が増加しているのに対し, CBCG 法は減少している．

表 5　ダルシーの流体問題 200^3 における反復回数

	反復回数
CG	3049
CBCG(10)	3050
CBCG(15)	3060
CBCG(20)	3060

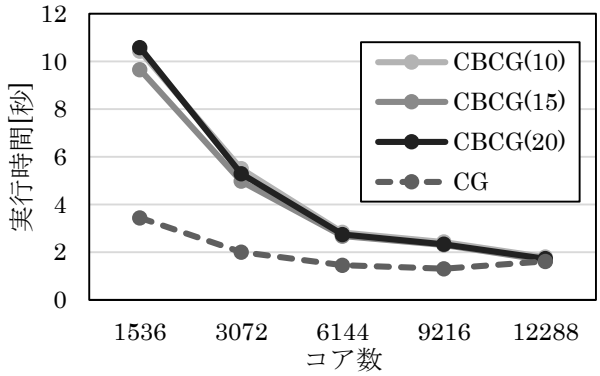

図 12　コア数増加による実行時間の推移 (問題サイズ 200^3)

9216 コアと 12288 コアでの実行時間の内訳を図 13，図 14 にそれぞれ示す．9126 コアでは，CG 法と CBCG 法の通信時間を見てみると，CBCG 法の方が多いことがわかる．CBCG 法の集団通信は CG 法が 2 回に対して$3/k$ 回であるため，集団通信による通信の増大ではないと考えられる．したがって，CBCG 法で$1/k$回分増えた隣接通信が原因だと考えられる．12288 コアでは，CBCG 法の通信時間が減少し，CG 法とほぼ同じ実行時間となった．また，9216 コアから 12288 コアにしたことで理論的に演算が 0.75 倍の時間となるはずである. 9216 コアでの CBCG 法$k = 20$の演算時間は約 1.3 秒で，12288 コアでは約 1.0 秒となり，約 0.77 倍に時間が削減されている結果となった．そのため，さらにコア数を増加させた場合に，演算時間も削減されることが見込まれ，CBCG 法がより少ない実行時間でとなることが考えられる．

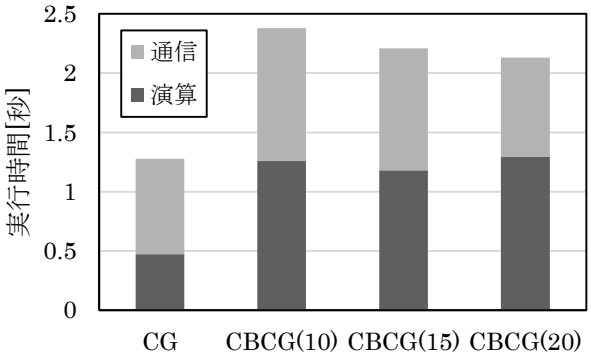

図 13　9216 コアでの実行時間の内訳

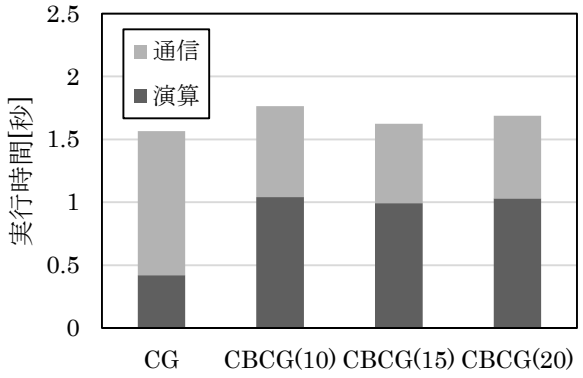

図 14　12288 コアでの実行時間の内訳

5.3.3 考察

今回の実験結果より，CG 法は高並列化に伴い通信時間が増加し，実行時間が増加した．それに対し，CBCG 法は集団通信の回数が削減されたことにより，高並列になるにしたがい，CG 法とほぼ同じ実行時間となった．また，実行時間の内訳によると，CG 法は通信時間が大半を占めるようになり，高並列環境において実行時間の削減が見込まれないと考えられる．CBCG 法では演算時間が CG 法よりも多くなってしまうが，通信時間は CG 法よりも少ない結果となった．よって，CBCG 法の集団通信を削減した効果は有効だと考えられる．

6. おわりに

本稿では，CBCG 法を高並列な環境で実験を行い，CG 法と比較し特性評価を行った．CBCG 法は集団通信の回数を削減している代わりに演算量が CG 法の約 3 倍となる．並列数が小さいときは全体の実行時間も CG 法の 3 倍程度かかる結果となった．並列数を上げると CG 法の実行時間が停滞するのに対し CBCG 法が CG 法とほぼ同じ実行時間となった．問題サイズ 100^3 においては 9216 コアで実行した場合に CBCG 法の実行時間が CG 法よりも少ない結果となり，集団通信の回数を削減した効果が出ていることがわかった．問題サイズ 200^3 においては，12288 コアで実行した場合に CBCG 法が CG 法とほぼ同じ実行時間となり，さらにコア数を増加させた場合に CBCG 法の方がより高速になることが考えられる．

今回使用した問題は規則構造だが，並列度を自由に変更させるために ParMETIS による領域分割をして計測を行った．FX10 ではトポロジーも指定でき，問題の構造とトポロジーを合わせることで，SpMV の性能は向上し，CBCG 法により有効だと考えられる．それを踏まえた上で，今後の課題として，隣接通信の最適化を施し，さらに通信による影響を減らす必要がある．また，前処理を施した CBCG 法での実験をする必要がある．

参考文献

[1] Hestenes, M.R. and Stiefel, E., Methods of Conjugate Gradients for Solving Linear Systems, Journal of Research of the National Bureau of Standards, Vol.49, No.6, pp.408-436 (1952).
[2] Chronopoulos, A. and Gear, C., s-step iterative methods for symmetric linear systems, Journal of Computational and Applied Mathmatics, Vol.25, pp.153-168 (1989).
[3] Toledo, S.A, Quantitative Performance Modeling of scientific Computations and Creating Locality in Numerical Algorithms, ph.D. thesis, Massachusetts Institute of Technology (1995).
[4] 本谷徹，須田礼仁，k 段飛ばし共役勾配法：通信を回避することで大規模並列計算で有効な対称正定値行列連立 1 次方程式の反復解法，情報処理学会研究報告，Vol.2012-HPC-133，NO.30 (2012).
[5] Hoemmen, M. F., Communication-avoiding Krylov subspace methods, ph.D. thesis, University of California Berkeley (2010).
[6] 須田礼仁，本谷徹，チェビシェフ基底共役勾配法，情報処理学会ハイパフォーマンスコンピューティングと計算科学シンポジウム(HPCS), 2013, pp72 (2013).
[7] 須田礼仁，李聡，島根浩平，数値的に安定性な通信削減クリロフ部分空間法，計算工学講演会論文集，Vol. 19 (2014).
[8] 東京大学情報基盤センタースーパーコンピューティング部門，FX10 スーパーコンピュータシステム(oakleaf-fx)，http://www.cc.u-tokyo.ac.jp/system/fx10/
[9] ParMETIS, http://glaros.dtc.umn.edu/gkhome/metis/parmetis/overview