[共同研究成果]

乱流大規模直接数値計算コードの SX-Aurora TSUBASA 上での性能評価

山本 義暢:山梨大学大学院総合研究部
 滝沢 寛之:東北大学サイバーサイエンスセンター
 加藤 季広:NEC・AIプラットフォーム事業部
 磯部 洋子:NEC・AIプラットフォーム事業部

乱流大規模直接数値計算コードの AOBA-A への移植及びその性能評価を行った。全対全通信 (all-to-all)通信を回避するコードにおいては、実際の高レイノルズ数直接数値計算条件のベンチマ ークにおいて、単体ベクトルエンジンにおいて理論演算性能の 9%、AOBA-A 全システム(576 ベ クトルエンジン)使用時に 57 Tflop/s の実行演算速度を得ることができた。前機種 SX-ACE 並びに Cray XC40 との比較においては、同一数のベクトルエンジン対ノードとの比較では約4倍程度の 計算速度向上が得られている。これにより AOBA-A で実行可能な条件では全ケースに対し1タイ ムステップ1秒程度での時間積分が可能となっていることを確認した。

1. はじめに

非線形・非定常の他自由度力学系の典型である乱流現象を理解する手法として、乱流の直接数 値計算(Direct Numerical Simulation, DNS)は、最も有力な手法である。しかし DNS においては流れ 場のレイノルズ数(Re = UL/v, U:代表速度、L:代表長、v:流体の動粘性係数)の 9/4 乗に比例する格 子分解能が必要となり、工学的に重要な高レイノルズ数を対象とすることが重要な課題となって いる。そのため高レイノルズ数乱流 DNS においては、スーパーコンピュータを用いた大規模容 量・高速計算が必須となる。周期境界条件が適用できる場合、フーリエ・スペクトル法が空間離 散化手法として最も高精度であり、同一精度を得るための計算コストにおいても最も高効率な手 法であるため、DNS における標準離散化手法として確立している。フーリエ・スペクトル法で必 要となる高速フーリエ変換(Fast Fourier Transform, FFT)は、計算方向に領域分割が施されている場 合、分割軸を変更して並列性を捻出する必要が生じる。この分割軸の変更は、全対全(all-to-all)通 信となるため、ネットワークバンド幅(特にフルバイセクションバンド幅)への要求が高くなる。

上記のように乱流 DNS においては、高メモリバンド幅・高速ネットワークシステムが要求され ベクトル並列計算機が最適システムとされてきた。第一筆者においても初代地球シミュレータ (SX-5 ベース) [1]以来、NEC SX シリーズ(SX-9 [2]、 SX-ACE [3])を対象とした DNS コード開発 [4], [5]を実施している。

ー方東北大学サイバーサイエンスセンターに 2020 年 10 月に導入された新ベクトル並列計算機 AOBA-A(NEC 製 SX-Aurora TSUBASA, 以降、SX-AT と表記) [6]は、高メモリバンド幅(ベクトル エンジンあたり 2.45Tflop/s の理論演算性能に対し、1.53TB/s のメモリアクセス速度)を維持してい るものの、初代地球シミュレータに代表される単段クロスバーネットワーク [1]のような特別な ネットワークシステムは有さず、x86/Linux であるベクトルホスト(VH)を通じた InfiniBand による ネットワーク構成(図 1 参照)となっており、その影響が懸念される。また新システムにおいては NEC Numeric Library Collection(NLC)による新数値計算ライブラリ [7]、ベクトルエンジン(VE)内 並列における OpenMP と自動並列の併用など多くの変更が試みられている。本研究ではベクトル 並列計算機上で開発された DNS コードを SX-AT に移植し、その性能評価を実施した。

2. 計算体系と計算手法の概要

2.1 計算体系と支配方程式

計算対象は図 2 に示す、一定圧力勾配により駆動される十分に発達したチャンネル流であり、 境界条件としては流れ方向(x)とスパン方向(y)に周期境界条件、壁面では no-slip 条件となる。流れ

— 1 —

場の支配方程式は非圧縮性流体の Navier-Stoke 方程式と連続式である。

2.2 計算手法 [4], [5]

 $\bullet k = 1$



図1 AOBA-A におけるネットワークシステム

図2計算対象(チャンネル流)と座標系

空間離散化手法としては、x,y方向にフーリエ・スペクトル法、壁垂直方向(z)に2次精度中心差分法を適用し、Fractional Step 法により Navier-Stokes 式と連続式をカップリングさせる。これにより派生する圧力ポアソン方程式(1)は、以下となる。

$$-\left(k_{x}^{2}+k_{y}^{2}\right)\Phi(l,m,k)+\left\{\frac{\Phi(l,m,k+1)-\Phi(l,m,k)}{dzw(k)}-\frac{\Phi(l,m,k+1)-\Phi(l,m,k)}{dzw(k-1)}\right\}\frac{1}{dz(k)}=D(l,m,k) \quad (1)$$

ここに k_x (= 2 $\pi l/L_x$, $l = 0, ..., N_x/2, N_x$: x 方向のフーリエモード数), k_y (= 2 $\pi m/L_y, m = -N_y/2, ..., N_y/2-1, N_y$: y 方向のフーリエモード数) は x, y 方向への波数, Φ , D は x, y 方向にフーリエ変換を施した圧力及び部分段階速度により評価した連続式にマイナスをかけタイムステップ幅で除した値, k(= 1, ..., N_z)は z 方向の格子点番号、 dz 及び dzw は壁垂直方向の格子間隔(不等間隔格子)を示す。式(1) は式(2)のように整理すると、各波数(l, m)に対して独立に $\Phi(k), \Phi(k+1), \Phi(k-1)$ に対する 3 重対角行列(Tri-Diagonal Matrix)となっている。

$$\left\{ \underbrace{\frac{1}{dz(k)} \left(\frac{-1}{dzw(k)} + \frac{-1}{dzw(k-1)} \right) - \left(k_x^2 + k_y^2\right)}_{A(l,m,k)} \Phi(l,m,k) = \underbrace{\frac{1}{dz(k) dzw(k)}}_{B(l,m,k)} \Phi(l,m,k+1) + \underbrace{\frac{1}{dz(k) dzw(k-1)}}_{C(l,m,k)} \Phi(l,m,k-1) + D(l,m,k) \right\}$$
(2)

式(2)の3 重対角行列は以下に示す TDMA(Tri-diagonal Matrix Algorithm)により各波数(*l*, *m*)に対して直接解くことができる。

(i) k=1,..., N_z(z 方向の格子点数)の時の P, Q を式(3)及び(4)により求める(forward step)

$$P(l,m,1) = \frac{B(l,m,1)}{A(l,m,1)}, Q(l,m,1) = \frac{D(l,m,1)}{A(l,m,1)}$$

$$\bullet k = 2, \dots, N_z$$

$$P(l,m,k) = \frac{B(l,m,k)}{A(l,m,k) - C(l,m,k)P(l,m,k-1)}$$

$$Q(l,m,k) = \frac{D(l,m,1) + C(l,m,k)Q(l,m,k-1)}{A(l,m,k) - C(l,m,k)P(l,m,k-1)}$$
(3)

(ii) $k = N_z, ..., 1$ に対し式(4)により順次Φが得られる(backward step)。 $\Box k = N_z$ $\Phi(l, m, N_z) = Q(l, m, N_z)$ $\Box k = N_z - 1, ..., 1$ $\Phi(l, m, k) = P(l, m, k)\Phi(l, m, k+1) + Q(l, m, k)$ (4)

またフーリエ・スペクトル法における非線形項の計算は擬スペクトル法を適用し、これにより派 生する aliasing error は 3/2 則により除去する。時間積分は圧力項に Euler 陰解法、その他の項に 2 次精度の Adams-Bashforth 法を用いる。

3. 並列化手法の概要

領域分割は図 3-(a)に示す壁垂直方向(z)への1次元領域分割を適用する。これより x, y 方向のフ ーリエ変換においては分割軸の変更は不要となる。一方圧力ポアソン方程式(2)の壁垂直方向(z)の TDMA においては、式(3), (4)は分割軸(z: k)に対して逐次計算となるため並列性が得られない。 (A) a2a コード

そこで一般には分割軸 z を変更し並列性を確保する。変更する分割軸に際しては、一般にチャンネル流における計算では、x, y, z 方向の格子点数が異なることから、単純に $z \rightarrow x$, あるいは $z \rightarrow y$ と変更すると分割数の不均一が生じるため、x, y 軸を 1 次元化(xy 軸)し、この xy 軸に対して 図 3-(b)のように領域分割を行う。これにより z(k)軸方向の TDMA においても完全かつ均一な並列 性を確保できる。しかしこの分割軸の変更には全対全(all-to-all)通信となりそのコストは高い(以下 本コードを a2a コードと呼ぶ)。



(a) domain decomposition for *z*-axis (*Nx*, *Ny*, *Nz/nprocs*)

(b) domain decomposition for xy-axis (*NxNy/nprocs*, *N*,)

図3 領域分割方法, MPI 並列数(nprocs) = 4 の場合

(B) shift $\exists - \vDash$

そこで all-to-all 通信を用いずに TDMA を図 4 のように計算する。図 4-(a)は x, y 方向の格子数を 3 及び 4 とした例であり、図中直方体の数字: 1,1 は x, y 方向の格子点番号を示す。従って一つの 直方体が z 方向への 1 回の TDMA 計算を意味する。この 1 回の TDMA 計算を以下の手順で計算 する(図 4-(b)参照)。

(1) まず壁垂直方向の第一格子点(k=1)を持つ rank 0 において式(3)を計算し、rank 0 に領域分割の 上限の P 及び Qの値を同期通信:mpi_sendrecv により rank 1 に送る。rank 1 が担当する k の値の範 囲において式(3)を計算し、同様に上限の P 及び Q の値を rank 2 に送る。これを rank *nprocs* -1(nprocs:MPI 並列数, 図 4 の例では*nprocs*= 4)まで繰り返す(forward step)。

(2) 次に $k = N_z$ を持つ rank nprocs -1 において式(4)を計算し、領域分割下限の Φ の値を同期通

- 3 -

信:mpi_sendrecv により rank *nprocs* – 2 に送る。これを rank 0 まで繰り返すことにより、k 方向の すべての格子点における圧力: Φ が求まる(backward step)。

但しこれは逐次計算となっており並列性はゼロである。そこで式(3)及び(4)は各波数(*l*, *m*)に対して は独立であることを利用して、図 5-(a)に示すように(*l*, *m*)に対して各ランク(MPI プロセス)の計算 をオーバーラーップさせることにより並列性を作る(本コードを shift コードと呼ぶ。なお図 5-(a) においては上記の同期通信:mpi_sendrecvのコストを省略している)。図 5-(a)において同一時間に おいて各ランクで重複計算となっている部分が並列性に相当する。一方 a2a コードにおいては図 5-(b)に示すように演算部分においては完全な並列性が得られるが、分割軸の転置コスト(all-to-all 通信)次第では shift コードが優位となることが期待できる。なお図 4 においては理解を容易とす るために *x*, *y* 方向 1 格子の *z* 方向への TDMA を計算単位としているが、実際の計算コードではベ クトル化及びベクトル長を確保するために複数の TDMA 計算をある程度まとめた単位(通常は *x* 方向の格子数分を1単位としている)を1回の計算としている。



⁽a) unit of TDMA for shift code, grid numbers: $(N_x/2+1, N_y, N_z) = (3, 4, N_z)$

- 4 --

(b) sequential TDMA calculation



図 4 shift コードにおける TDMA の逐次計算

図 5 shift コード及び a2a コードにおける並列性,格子数(N_x/2 + 1, N_y, N_z) =(3, 4, N_z), MPI 並列 数:*nprocs* = 4 の場合、図 4-(a)と同様に 1,1 等は x, y 方向の格子点番号を示す。 (a) shift コード、(b)a2a コード

4. AOBA-A (SX-Aurora TSUBASA)への移植作業

上記 DNS コードを SX-AT に移植するあたり以下の作業が必要となった。

(1) FFT ライブラリの変更: 本 DNS コードは、1 物理量に対し*x*, *y* 方向への 2 次元実フーリエ変 換/逆変換(2D-FFT)を科学技術計算ライブラリ:ASL(逐次版) [8]により *z* 方向格子数回実施す る。ノード内に自動並列を使用する場合(ハイブリッド並列)は、z 方向を分割し 2D-FFT 部分を 並列実行していた。しかし SX-AT においては ASL の 2D-FFT(逐次版)はスレッドセーフではな いため、ハイブリッド並列が不可となる。そこで NEC Numeric Library Collection の FFTW3 インタ ーフェース [9]への変更及び *z* 方向へ OpenMP を利用してのハイブリッド並列化を行った。

(2) 自動並列不可部分の OpenMP 化: 従来の SX シリーズでは OpenMP と自動並列の併用が不可 であったが、SX-AT においてはこの併用が可能となった。そこで自動並列不可部分に対し OpenMP による並列化を実施した。

(3) all-to-all 通信方法の変更: 従来の SX シリーズにおいては、グローバルメモリ機能を活用した mpi_put 関数による高速ノード間通信 [10]が可能であり all-to-all 通信においてはこれを利用して いた。一方 SX-AT ではグローバルメモリ機能が廃止されている。そこで mpi_alltoallv への変更を 行った。

(4) その他:この他の変更点としてはコンパイルオプションの変更、基準エンディアンが little endian となったことによる対応、ローカル変数がスタック領域に割り付けられることになったことにより、スタックサイズの指定(本 DNS コードの場合、デフォルトではスタック領域が不足する)等が必要となった。

5. AOBA-A (SX-Aurora TSUBASA)上での性能評価

5.1 評価条件

ベンチマークにおいては、高レイノルズ数効果解析条件 [11],[12]を用いた。表1にその条件を まとめる。ここに Re_r(= u_rh/v , u_r :壁面摩擦速度,h:チャンネル半値幅,v:流体の動粘性係数)は摩擦 レイノルズ数であり、 N_x , N_y はx,y方向へのフーリエモード数, N_z は壁垂直方向への格子点数を示 す。なお物理空間における非線形項計算時はx,y方向にそれぞれ $3N_x/2$, $3N_y/2$ 格子を用いて計算を 行う(3/2 則)。 Δx^+ , Δy^+ , Δz^+ はx,y,z方向への空間解像度を示し、上付き添え字+は壁面摩擦速度と 動粘性係数により無次元化された値であることを示す。各ケースにおいて 100 タイムステップの 時間積分を行い、初期設定及び乱流統計処理を含んだ結果である(但し瞬間乱流場の IO は除く)。

Reτ	L_x/h	L_y/h	N_x	N_y	N_z	Δx^+	Δy^+	Δz^+
500	16.0	6.4	500	384	384	16.0	8.3	0.4-5.3
1000	16.0	6.4	1000	768	512	16.0	8.3	0.6-8.0
2000	16.0	6.4	2000	1536	1024	16.0	8.3	0.6-8.0
4000	16.0	6.4	4000	3072	2048	16.0	8.3	0.6-8.0
8000	16.0	6.4	6912	5760	4096	18.5	8.9	0.6-8.0

表1 ベンチマーク条件 [11], [12]

5.2 ベンチマーク結果

以下に示す実行時間(elapsed time)[s]、実行演算速度[Gflop/s]等は、SX-AT における性能情報解 析ツール(FTRACE)により取得した値である。

5.2.1 単一ベクトルエンジン(VE)性能

図 6-(a)に単一 VE(総 core 数:8)における Re_r=500 のケースの実行演算速度[Gflop/s]を示す。ここ で#AP は VE 内スレッド並列数を示し、#MPI(=8/#AP)は MPI 並列数を示す。a2a コード及び shift コード両者とも#AP = 1(#MPI = 8, flat MPI)のケースが最も効率がよく、shift コードにおいては理論演算性能の 8.5%の演算速度が得られている。#MPI を減らし、#AP を増やすと演算速度はやや減少し shift コードの#AP = 4(#MPI = 2)の時は理論演算性能の 7.7%の実行効率となっている。a2a コードと shift コードの比較においては、flat MPI の場合はほぼ同じ演算速度が得られているものの、#AP が増加すると shift コードの方が高速である。#AP=1(#MPI=8)の shift コードにおいては 1 タイムステップあたりの実行時間は約 0.40 秒である。

5.2.2 単一ベクトルホスト(VH)性能

図 6-(b)に単一ベクトルホスト(VH)内における Re_r= 1000 のケースの実行演算速度[Gflop/s]を示 す。ここで#VE はベクトルエンジン数(最大 8)を示し、ベクトルエンジン内においては MPI 並列 数 = 2, #AP = 4 と固定している。従って、総 MPI 並列数は、#VE × 2 となる。shift コードにお いては#VE = 2 の時に理論演算性能の 8.9%, #VE = 8 の時に 7.0%の実行効率が得られており、これ は単一 VE 内効率と同等である。a2a コードと shift コードの比較においては、shift コードの優位 性がより顕著となっている。#VE = 2,8 の比較により算出した並列化効率は、shift コードが 78.8%, a2a コードが 64.2%となっている。#VE = 8 の shift コードにおいては 1 タイムステップあたりの実 行時間は約 0.33 秒である。

5.2.3 マルチ VH 性能

図 6-(c)にマルチベクトルホストにおける Re_r=2000 のケースの実行演算速度[Gflop/s]を示す。こ こで#VH はベクトルホスト数を示し、1 ベクトルホスト内の VE 数は 8 であり、1VE 内の MPI 並 列数を 2, スレッド並列数を 4 と固定している。従って総 MPI 並列数は、#VH×16 となる。なお a2a コードは、#VH = 2 ではメモリオーバとなり動作しなかった。shift コードにおいては#VH = 2 の場合の理論演算性能に対する実行効率は 7.0%であり、#VH = 8 の場合の実行効率は 4.3%とな り、#VH=2,8 の比較により算出した並列化効率は 70.9%となり、単一 VH の場合と比べて、実行 効率、並列化効率ともに減少している。#VH = 8 においては 1 タイムステップあたりの実行時間 は約 0.59 秒である。shift コードと a2a コードの比較においては、shift コードの方が約 3 倍程度高 速である。

 $Re_{\tau} = 4000 \text{ mbor}$ ースはメモリ量の制約により、AOBA-A における最大ジョブクラス(#VH = 32, #VE = 256, #core = 2048)のみで動作する。上記と同様にベクトルエンジン内においては MPI 並列数 = 2, #AP = 4 としている。表 2 に計測結果を示す。なお shift コードにおいては前機種 SX-ACE/256 ノードの結果も併記している(SX-ACE の場合#VE はノード数に相当)。本条件においても a2a コードに対する shift コードの優位性が確認できる。SX-ACE との比較においては、実行効率において 3 分の 1 程度に低下しているものの、約 3.3 倍の実行演算速度(20.92TFlop/s)が得られている。この 20.92[Tflop/s]は全機種 SX-ACE の最大ジョブクラス 1024 ノードでの Re_{τ} =8000 のケースは動作しない。

AOBA-A の全システム(#VH = 72, #VE = 576)を用いて実施した shift コード: 3200×5120×4608 格子条件においては、実行演算速度: 57 Tflop/s, 実行効率:4%が得られている。

5.2.4 他システムでの性能

図 6-(d)に京都大学学術情報メディアセンターの Cray XC40 [13]の最大 8 ノード(ノードあたり理 論演算性能:3.05Tflop/s, core 数:68,メモリ量:85GB)を用いた Re_τ=1000 のケースの結果を示す。 shift コードにおいては2ノードにおいて、理論演算性能に対する実行効率1.5%、8ノードにおいて1.4%であり、ノード数2,8の比較により算出した並列化効率は91.3%である。a2a コードにおいては shift コードと比較し、実行演算速度で10%程度劣っているが、ノード数2,8の比較により 算出した並列化効率は90.2%であり高い効率を維持している。shift コードノード数8の場合、1タ イムステップの実行時間は約1.38秒であり、Cray XC40/8ノード(理論演算性能:24.4Tflop/s)に対 し、約80%の理論演算性能である AOBA-A #VE = 8(理論演算性能:19.6Tflop/s)の方が約4.2 倍高 速である。



図 6 a2a コード、shift コードの性能評価:実行演算速度[Gflop/s]

Ver.	Re_{τ}	computer	#VE	#MPI/VE	#AP/VE	#step	elapse[s]	Tflop/s	実行効率
a2a	4000	SX-AT	256	2	4	100	313.5	11.06	1.76%
shift	4000	SX-AT	256	2	4	100	165.8	20.92	3.33%
shift	4000	SX-ACE	256	1	4	100	550.6	6.30	9.62%
shift	8000	SX-ACE	1024	1	4	100	1027.6	21.02	8.02%

表 2 AOBA-A 最大ジョブクラスでの性能:ケース Re_t = 4000 (参考値:Re_t=8000)

5.2.5 性能評価に対する考察

表 3 に a2a コードにより算出した TDMA の演算及び通信コストを示す。ここに type は MPI 通信のハード上での違いを表しており、図1に示すように使用する VE 数により、type 1 はメモリコピー、type 2 は Direct Memory Access (DMA), type 3 は VH を介した DMA、type 4 は VH を介した

- 7 --

InfiniBand による通信となることを示している。すべてのケースにおいて TDMA の演算量(C_{TDMA}) は DNS コード全体の演算量(C_{total})に対して 1%程度となっている。一方図 5-(b)に示した TDMA 演算部と all-to-all 通信部の計算時間比(T_{a2a}/T_{TDMA})は、通信がメモリコピーで行われる type 1 で 2.2 倍であり、演算時間よりも all-to-all 通信にかかる時間が大きいことがわかる。また DMA 通信の type 2 が 5.4 倍、VH を介した DMA 通信の type 3 が 20.6 倍、VH を介した InfiniBand 通信の type 4 が 30-100 倍となり通信タイプの違いにより all-to-all 通信に要する時間が圧倒的に大きくなることが確認できる。実際全計算時間(T_{total})に占める TDMA 演算及び all-to-all 通信にかかる時間 ($T_{\text{TDMA}}+T_{a2a}$)は type 4 で 40% 以上に及ぶ。all-to-all 通信において、総送受信量を要した時間で除して求めた実効バンド幅[GB/s]は通信 type に応じて大きく変化していることがわかる。ベクトルホスト数を増やすことにより実効バンド幅は向上しているもののその比率は一定ではなく、演算速度の向上に比して鈍化している。

以上より SX-AT においては all-to-all 通信と演算をオーバーラップして隠ぺいする [14]か、all-to-all 通信自体を回避する方策をとる必要があることがわかる。しかし TDMA については演算量 自体が全体演算量の 1%程度しかなく、実行時間においても all-to-all 通信の比率が圧倒的に大きい ことにより、演算とのオーバーラップによる隠ぺい効果は期待できない。従って shift コードのよ うな all-to-all 通信を回避する対策が不可欠であると言える。

type*	Re_{τ}	#VE	C_{total}	C_{TDMA}	C_{TDMA}	T_{a2a}	$(T_{\text{TDMA}} + T_{a2a})$	実効バンド		
			[Gflop/step]	[Gflop/step]	$/C_{\text{total}}$ [%]	$/T_{\text{TDMA}}$	$/T_{\text{total}}$ [%]	幅[GB/s]		
1	500	1	82.36	1.14	1.39	2.2	4.2	190.3		
2	1000	2	450.75	6.09	1.35	5.4	16.8	64.8		
3	1000	8	450.76	6.09	1.35	20.6	36.4	67.9		
4	2000	32	3975.62	48.76	1.23	29.4	40.7	189.9		
4	4000	256	34688.06	390.07	1.12	98.0	46.9	515.5		

表 3 DNS コードにおける TDMA 及び all-to-all 通信のコスト(a2a コード)

*1:memory copy, 2: DMA(Direct Memory Access), 3: DMA via VH, 4:Infiniband via VH (図 1 参照)



図 7 a2a コードと shift コードの TDMA 計算時間比

図 7 に各ケースにおける a2a コード及び shift コードの TDMA に要する時間の比を示す(a2a コード場合は、表 3 の $T_{TDMA}+T_{a2a}$ の値)。shift コードは単一 VE 時以外においては a2a コードに対して 2 倍以上の優位性を持っているものの、その優位性は一律ではないことがわかる。これは shift

コードにおいては、z方向分割数が減少すると図4-(a)に示した並列性が減少することに起因する。 一方、all-to-all 通信はバイセクションバンド幅(実効バンド幅)に依存するが、表3に示すように VE 数の増加に伴い実効バンド幅自体は向上していることにより、VE 数の増加に伴う shift コード の a2a コードに対する優位性が複雑に変化するものと思われる。

6. まとめ

本研究では、2020 年 10 月に導入された新ベクトル並列計算機 AOBA-A(NEC・SX-Aurora TSUBASA)上への大規模乱流 DNS コードの移植を行い、その性能評価を実施した。その結果、

(1) SX-Aurora TSUBASA は同一 VE 数と前機種(SX-ACE)及び他機種 Cray XC40 の同一ノード数比較において、約4倍程度の高速演算が可能であることを確認した。

(2) AOBA-A 全システム(576VE)を用いた場合は 57Tflop/s(理論演算性能の4%)の実行演算速度が得られた。

(3) 以上により AOBA-A 上で実行可能な直接数値計算を1タイムステップ1秒程度での実行が可能となっている。

(4) 一方、近年の並列計算機は演算性能の向上に比して、ノード間(SX-AT では VE 間あるいは VH 間)通信速度の向上は鈍化しており、SX-AT においても all-to-all 通信を用いる手法は VE 内以外においては性能劣化をもたらし、これを回避する対策をとる必要がある。

今後は本評価結果を踏まえ、AOBA-A の 20 倍規模の拡張が予定されている次期システムをタ ーゲットとした大容量・高速 DNS コードの開発を行う予定である。

謝辞

本研究は東北大学サイバーサイエンスセンターとの共同研究により実施した。AOBA-A 全シス テム(576VE)を利用した計算は、大規模並列ジョブチャレンジ制度を利用して行った。また本研究 は栢森情報科学振興財団による研究助成を受けて遂行した。記して謝意を表する。

参照文献

- [1] "地球シミュレータ システムの概要(ES)," JAMSTEC, [オンライン]. Available: http://www.jamstec.go.jp/es/jp/es1/system/hardware.html. [アクセス日: 16 5 2021].
- [2] 大泉健治, 江川隆輔, 伊藤英一, 岡部公起, 小林広明, "スーパーコンピュータシステム SX-9の紹介と性能評価," *SENAC*, Vol.42, No.1, pp. 129-135, 2009.
- [3] 山下毅, 森谷友映, 佐々木大輔, 齋藤敦子, 小野敏, 大泉健治, "スーパーコンピュータシス テム SX-ACE の紹介," SENAC, Vol. 48, No.1, pp. 39-46, 2015.
- [4] Y. Yamamoto and T. Kunugi, "Direct numerical simulation of a high-Froude-number turbulent open-channel flow," *Phys. Fluids*, vol. 23, no. 12, p. 125108, 2011.
- [5] Y. Yamamoto and T. Kunugi, "MHD effects on turbulent dissipation process in channel flows with an imposed wall-normal magnetic field," *Fusion Eng. Des.*, vol. 109, no. 111, pp. 1137-1142, 2016.
- [6] 山下毅, 森谷友映, 佐々木大輔, 齋藤敦子, 小野敏, 大泉健治, 滝沢寛之, "東北大学サイバ ーサイエンスセンター スーパーコンピュータ AOBA の紹介," SENAC, Vol.No.54, No.1, pp. 50-55, 2021.
- [7] "NEC Numeric Library Collection 2.3.0 ユーザーズガイド," NEC Corporation, [オンライン]. Available: https://www.hpc.nec/documents/sdk/SDK_NLC/UsersGuide/main/ja/index.html. [アクセス日: 16 5 2021].

- [8] "ASL ネイティブインタフェース," NEC Corporation, [オンライン]. Available: https://www.hpc.nec/documents/sdk/SDK_NLC/UsersGuide/asl/f/ja/index.html. [アクセ ス日: 16 5 2021].
- [9] "FFTW3 インタフェース (Fortran 用)," NEC Corporation, [オンライン]. Available: https://www.hpc.nec/documents/sdk/SDK_NLC/UsersGuide/fftwint/f/ja/index.html. [アク セス日:1652021].
- [10] 小林広明, 江川隆輔, 岡部公起, 伊藤英一, 大泉健治, "東北大学サイバーサイエンスセン ターの取り組みと SX-9 の性能評価," [オンライン]. Available: https://jpn.nec.com/hpc/tech/journal-sx9/080413.html. [アクセス日: 16 5 2021].
- [11] Y. Yamamoto and Y. Tsuji, "Numerical evidence of logarithmic regions in channel flow at Ret= 8000," *Phys. Rev. Fluids*, vol. 3, no. 1, p. 012602, 2018.
- [12] Y. Kaneda and Y. Yamamoto, "Velocity gradient statistics in turbulent shear flow -an Extension of Kolmogorov's local equilibrium," J. Fluid Mech., submitted.
- [13] "スーパーコンピュータシステム," 京都大学学術情報メディアセンター, [オンライン].
 Available: http://www.iimc.kyoto-u.ac.jp/ja/services/comp/supercomputer/. [アクセス日: 1652021].
- [14] S. Maeyama, T. Watanabe, Y. Idomura, M. Nakata, M. Nunami and A. Ishizawa, "Computation-Communication Overlap Techniques for Parallel Spectral Calculations in Gyrokinetic Vlasov Simulations," *Plasma and Fusion Research*, vol. 8, p. 1403150, 2013.