Vol.2016-HPC-156 No.10
2016/9/16

データ解析・シミュレーション融合スーパーコンピュータシステム Reedbush-U の性能評価

塙 敏博[1,a)]　中島 研吾[1]　大島 聡史[1]　伊田 明弘[1]　星野 哲也[1]　田浦 健次朗[1]

概要：東京大学情報基盤センターでは，データ解析・シミュレーション融合スーパーコンピュータシステム Reedbush を導入し，2017 年 3 月より全系稼働開始予定である．Reedbush システムは、Intel Xeon E5 (Broadwell-EP) プロセッサに加えて NVIDIA Tesla P100 (Pascal) GPU を一部計算ノードに搭載する他，高速ファイルキャッシュシステムや，InfiniBand EDR などを始めとして導入時点で最新の技術を集めたシステムである．本稿では 2016 年 7 月から稼働を開始した汎用 CPU のみからなる Reedbush-U サブシステムの性能について報告する．

1. はじめに

東京大学情報基盤センター（以下，当センター）ではこれまで数年間にわたって Yayoi (日立 SR16000, IBM Power7, 54.9 TFLOPS), Oakleaf-FX (富士通 PRIMEHPC FX10, SPARC IXfx, 1.13 PFLOPS), Oakbridge-FX (Oakleaf-FX と同モデル, 136.2 TFLOPS) の 3 機種のスーパーコンピュータシステムを運用してきた．これらのシステムは，工学，地球・宇宙科学，材料科学といった様々な分野の，2,000 を超える利用者に活用されている．利用者の半数以上が学外からの利用者であり，また最近では生物学，生体力学，生化学などの分野の利用者が増加している．このような計算機利用需要の増加から，Oakleaf/Oakbridge-FX は大変混雑しているのが現状である．特に 2015 年のシステム利用率は 80%以上に及び，計算資源の拡充が急務であった．

これまで当センターは，筑波大学と共同で最先端共同 HPC 基盤施設 (JCAHPC: Joint Center for Advanced High Performance Computing[1]) を立ち上げ，Post T2K システムの調達を行ってきた．しかしながら，Post T2K システムは当初の導入予定から遅れたことや，また最新アーキテクチャであることから，ユーザがプログラムの調整を行いプロダクトラン可能になるまでには，やや時間を要する可能性があった．

一方，Oakleaf-FX は 2018 年 3 月に運用を終える予定であり，2018 年の秋頃より新しいシステム (Post FX10) の運用開始を目指している．現在のシステムは主として計算科学や工学向けに利用されており，また FX10 システムもそのような目的の利用を想定し設計されている．Post FX10 システムにおいては，ビッグデータ解析や人工知能といった，近年盛り上がりを見せている新たな分野の要求をも満たすシステムの開発を目指している．

これらの現状を踏まえて，当センターでは新たに「データ解析・シミュレーション融合スーパーコンピュータシステム」を導入することにした．本システムでは計算ノードの一部に GPU が搭載される．当センターで演算アクセラレータを搭載したシステムを導入するのは今回が初めてである．以前は，GPU のプログラムには，CUDA のような専用のプログラミング言語を用いて記述する必要があった．そのため，多数のユーザに，そのような言語を習得してもらうのは困難であると考えてきた．しかし近年，OpenACC といった指示文ベースのアクセラレータ用並列プログラミング言語が標準的に使われるようになり，実用に耐えうる十分な性能が得られるようになってきた．さらに，データ科学や機械学習など，従来のユーザとは異なる分野からも，GPU 搭載スパコンへの期待やニーズが高まっている．従って，本システムには

(1) Oakleaf/Oakbridge-FX の混雑緩和
(2) Post FX10 システムに向けてのテストベッドシステム

の 2 つの大きな役割が期待されている．

データ解析・シミュレーション融合スーパーコンピュータシステムは，2016 年 7 月より一部稼働を開始し，「Reedbush システム」の愛称で呼ばれている．本稿では，稼働を開始した Reedbush-U サブシステムを用いて性能評価を行った結果を報告する．

なお，Reedbush システムは，現在試験運転期間中であり，性能向上・安定のため，ドライバやソフトウェアなど

[1] 東京大学 情報基盤センター
[a)] hanawa@cc.u-tokyo.ac.jp

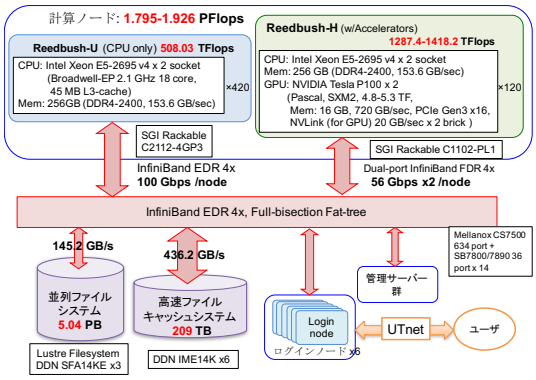

図 1　Reedbush システムの概要

図 2　Reedbush システムの外観

の更新を頻繁に行っている．従って，本稿と本サービス開始後の実システムとでは性能が一部異なる可能性がある．

2. Reedbush システムの紹介

2.1 概要

Reedbush システムは、**図 1** に示すように、CPU のみのノードからなる **Reedbush-U** と、演算アクセラレータとして GPU を搭載したノードからなる **Reedbush-H** の 2 つのサブシステムから構成され、それぞれは独立のシステムとして運用される．計算ノードに搭載されるプロセッサや GPU については次節以降で詳しく述べるが、2016 年 4 月時点で最新の製品である．さらに、計算ノード間インタコネクトには InfiniBand EDR、ストレージとして Lustre ファイルシステムに加えて高速ファイルキャッシュシステムを採用するなど、導入決定時点での未発表製品を含む最新製品を採用している．これらコモディティ技術をベースにした最新製品を導入することにより、運用期間中のハードウェアの陳腐化を抑制する一方で、運用に向けた準備にはこれまでのソフトウェア資産や経験を活かすことができる．

図 2 に Reedbush システムの外観を示す．また、**表 1** にシステム全体の仕様を示す．システム全体は、フルバイセクションバンド幅を持つ 1 つの Fat-tree 網として構成され、ノード当たり 100 Gbps を超える InfiniBand ネットワークにより接続される．

Reedbush システムは空冷システムであり、消費電力は 368.4 kW(冷却除く) の見込みである．

表 1　Reedbush システム全体仕様

Reedbush-U	総理論演算性能	508.03 TFLOPS
	総ノード数	420
	総主記憶容量	105 TByte
	インタコネクト I/F	InfiniBand EDR 4x 1 リンク / ノード (100 Gbps)
Reedbush-H	総理論演算性能	1297.15〜1417.15 TFLOPS (うち GPU: 1152.0〜1272.0 TFLOPS)
	総ノード数	120
	総主記憶容量	30 TByte
	インタコネクト I/F	InfiniBand FDR 4x 2 リンク / ノード (56 Gbps x2)
ノード間インタコネクト		InfiniBand EDR 4x フルバイセクションバンド幅 Fat-tree
並列ファイルシステム	種類	Lustre ファイルシステム DDN SFA14KE
	サーバ (OSS)	
	サーバ (OSS) 数	3 セット (6 ノード、12 サーバ)
	ストレージ容量	5.04 PByte
	バンド幅	145.2 GB/秒
高速ファイルキャッシュシステム	サーバ	DDN IME14K
	サーバ数	6 セット (12 ノード)
	容量	209 TByte
	バンド幅	436.2 GB/秒

2.2 汎用計算サブシステム：Reedbush-U

各計算ノードは、**表 2** に示すように、各ソケットに 18 コアの Intel Xeon E5 プロセッサ (開発コード名: Broadwell-EP) を 2 ソケット搭載し、256 GB の DDR4 メモリを搭載する．ノードあたり性能は 1.2 TFLOPS、メモリバンド幅は 153.6 GB/sec である．

表 1 に示すように、Reedbush-U サブシステム全体は 420 台の計算ノードからなり、各ノードは 100 Gbps の InfiniBand EDR によりフルバイセクションバンド幅を持つ Fat-tree トポロジで接続されている．ピーク演算性能は 508.03 TFLOPS, 総メモリ容量は 105 TByte である．

2.3 演算加速サブシステム：Reedbush-H

各計算ノードは、表 2 に示すように、Reedbush-U と同じ CPU、メモリを搭載しており、加えて**表 3** に示す通り、2 基の NVIDIA Tesla P100 GPU (開発コード名: Pascal) を搭載する．この GPU は 1 基あたり、4.8〜5.3 TFLOPS と極めて高い性能を持ち、また 16 GByte の HBM2 (High Bandwidth Memory) を搭載し、メモリバンド幅は 720 GByte/秒に達する [2].

特徴的なのは、**図 3** に示すように、

- 新しい高速インタコネクトである NVLink により 2 基の GPU 間が 40 GByte/秒のバンド幅で接続されていること
- 各 GPU に近接した InfiniBand FDR の HCA (Host Channel Adapter) が用意され、GPU メモリの内容を他のノードとの間で直接送受信できるように工夫されていること

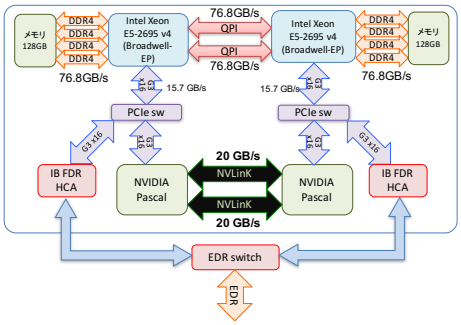

図 3　Reedbush-H ノードの構成

である．

表 1 に示すように、Reedbush-H サブシステム全体は 120 台の計算ノードからなり、各ノードは 56 Gbps の Infini-Band FDR を 2 リンク持ち、ノード当たりでは 112 Gbps を超えるフルバイセクションバンド幅を持つ Fat-tree トポロジで接続されている．ピーク演算性能は 1287.4〜1418.2 TFLOPS、総メモリ容量は 30 TByte である．

2.4　ストレージ

Reedbush は、ストレージとして 5.04 PByte の並列ファイルシステムと、209 TByte の高速ファイルキャッシュシステムを備える．並列ファイルシステムは Lustre ファイルシステムであり、実際にデータを格納する OSS (Object Storage Server) として DataDirect Networks 社の SFA14KE を 3 セット備え、計算ノード群に対して合計 145.2 GB/秒のバンド幅を提供する．高速ファイルキャッシュシステムには、同じく DataDirect Networks 社の IME14K を 6 セット用いる．これは、多くの SSD を搭載した複数のサーバを用いて、ファイル書き込みをまとめて高速化するバーストバッファや、ファイル読み込みのキャッシュなどといった機能を実現するもので、計算ノード群に対して合計 436.2 GB/ 秒のバンド幅を実現する．

並列ファイルシステムについては、RAID6 によるディスク冗長化、ファイルサーバやコントローラの二重化などにより、信頼性・可用性・耐故障性を高めていると同時に、無停電電源装置 (UPS) を備え、万一の停電に備えてバッテリバックアップを行っている．

高速ファイルキャッシュシステムについては、erasure coding という技術で、IME を構成するサーバ間でパリティデータを保持することで、高いバンド幅と高い信頼性を両立させている．

3.　ベンチマークによる性能評価

本章では，計算ノードに対するベンチマークとして，STREAM, HPL, HPCG, インタコネクトおよび MPI ライブラリのベンチマークとして Intel MPI Benchmark, ファイルシステムに対するベンチマークとして mdtest, IOR の結果を示す．さらに次章では，より実アプリケーションに近い Poisson 3D, GeoFEM/Cube 行列生成部を用いて評価を行う．また，GeoFEM/Cube 反復法ソルバーを用いて，MPI ライブラリの違いによる性能比較を行う．

表 2　計算ノードの仕様 (Reedbush-U, H 共通)

CPU	プロセッサ	Intel Xeon E5-2695v4 (Broadwell-EP) × 2 ソケット
	周波数・コア数	2.1 GHz, 18 コア × 2 ソケット
	ピーク性能	1209.6 GFLOPS
メモリ	種別・構成	DDR4-2400, 4 チャネル × 2 ソケット
	容量	256 GB
	バンド幅	153.6 GB/秒

表 3　Reedbush-H 計算ノードの GPU 仕様

プロセッサ		NVIDIA Tesla P100 (Pascal)
搭載数		2 基
GPU 間接続		NVlink × 2 brick (40 GB/sec)
CPU-GPU 間接続		PCI Express Gen3 x16 レーン (16 GB/sec)
GPU 単体	演算ユニット	56 SM (Symmetric Multiprocessor) × 64 CUDA コア (単精度), 32 CUDA コア (倍精度)
	ピーク性能	4.8〜5.3 TFLOPS
	メモリ種別	HBM2
	メモリ容量	16 GByte
	メモリバンド幅	720 GByte/秒

3.1　STREAM ベンチマーク

STREAM ベンチマーク [3], [4] を用いて計算ノードのメモリバンド幅を測定した．STREAM ベンチマークでは，

Copy:　配列のコピー $a(i) = b(i)$

Scale:　配列のスカラ倍 $a(i) = q \times b(i)$

Add:　2 つの配列の加算 $a(i) = b(i) + c(i)$

Triad:　Scale と Add の組み合わせ $a(i) = b(i) + q \times c(i)$

の 4 種を測定することができる．

ここでは，1 計算ノード中の 2 ソケットに均等に複数の OpenMP スレッドを配置した場合を測定した．C 版のプログラムを用い，Intel C Compiler 16.0.3 を用いてコンパイルした．コンパイルオプションには `-xCORE-AVX2 -O3 -qopt-streaming-stores always -DSTREAM_ARRAY_SIZE=400000000` を指定した．また，実行時には，2 ソケットに均等にプロセスを配置するため，環境変数 `KMP_AFFINITY=scatter` を指定した．

結果を図 4 に示す．Copy, Scale については，12 コア（ソケット当たり 6 コア）でバンド幅が飽和し，それぞれ 120.6, 120.1 GB/sec であり，このとき理論ピーク値と比較してそれぞれ 78.5, 78.2%である．一方，Add, Triad については，それぞれ 28 コア，24 コアの場合にメモリバンド幅が 130.5, 130.4 GB/sec と最大値を示した．これらは理論ピーク比では 85.0, 84.9%である．なお，FX10 では Triad 64.7 GB/sec, 理論ピーク比 74.3%であり [5]，Reedbush 1 ソケットとほぼ同等のメモリバンド幅であることがわかる．

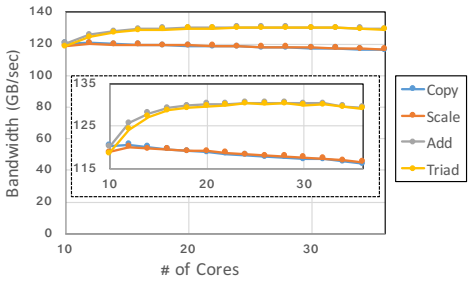

図 4　STREAM ベンチマークの結果

3.2　HPL ベンチマーク

LU 分解による連立一次方程式の求解を行うベンチマーク [6] であり，倍精度行列積演算 (Level-3 BLAS DGEMM) の性能がベンチマーク結果に大きな影響を与えることが知られている．

Intel MKL(Math Kernel Library) 11.3.3 に含まれる HPL 2.1 対応バイナリ (`xhpl_intel64_dynamic`) を使用した．また，MPI ライブラリには，Intel MPI 5.1.3 を用いた．

3.2.1　ノード単体性能

計算ノード 1 ノードにおいて，2 MPI プロセス × 18 スレッドを用い，MPI プロセスを各ソケットにそれぞれ割り当てた．問題パラメータ (HPL.dat) は以下の通り設定した．

- $N = 98112$, $NB = 192$, $P = 1$, $Q = 2$

実行の結果，1149.6 GFLOPS の性能を得た．これは理論ピーク性能の 95.0%である．

3.2.2　Reedbush-U 全系での性能

計算ノード 420 ノードを用い，1 ノードあたり 1 MPI プロセス ×36 スレッドとして実行した．問題パラメータは以下の通り設定した．

- $N = 1538688$, $NB = 192$, $P = 20$, $Q = 21$

実行の結果，446.3 TFLOPS の性能を得た．これは理論ピーク性能の 87.8%である．

3.3　HPCG ベンチマーク

HPCG は，HPC システムのための，より実アプリケーションに近いベンチマークとして提案されているもので，有限要素法から得られる疎行列を対象として共役勾配法 (Conjugate Gradient, CG 法) を用いて連立一次方程式を解く部分の演算性能を求めるものである．[7]

Intel MKL(Math Kernel Library) 11.3.3 に含まれる HPCG 3.0 対応バイナリ (`xhpcg_avx2`) を使用した．

3.3.1　ノード単体性能

計算ノード 1 ノードにおいて，2 MPI プロセス × 18 スレッドを用い，MPI プロセスを各ソケットにそれぞれ割り当てた．MPI ライブラリは，Intel MPI 5.1.3 を用いた．問題パラメータ (hpcg.dat) は以下の通り設定した．

- $nx = ny = nz = 144$

実行の結果，21.9 GFLOPS の性能を得た．これは理論ピーク性能の 3.6%である．（但し，本実行は 104.7 秒しか実施していない．）

3.3.2　Reedbush-U 全系での性能

計算ノード 420 ノードを用い，1 ノードあたり 2 MPI プロセス ×18 スレッドとして実行した．問題パラメータはノード単体の場合と同じである．

MPI ライブラリには，Intel MPI のための wrapper ライブラリ perfboost 1.14 を介して，SGI MPT 2.14 を用いた．今回は Intel MPI を用いたコンパイル済みバイナリを用いたため，wrapper ライブラリを用いる必要があった．

実行の結果，8457.78 GFLOPS の性能を得た．これは理論ピーク性能の 1.66%である．

HPCG については，MPI 通信の影響が大きいため，次に述べる Intel MPI benchmark の評価結果に基づいて最適なライブラリと実行パラメータを選ぶことで改善できる可能性がある．

3.4　Intel MPI ベンチマーク

ノード間インタコネクトである InfiniBand EDR の性能を測定するため，Intel MPI ベンチマーク (IMB) 4.1.1[8] を用いた．

3.4.1　MPI ライブラリ

Reedbush システムは，ノード間インタコネクトにコモディティ製品を用いたシステムであり，MPI ライブラリとしてオープンソースを含む複数のライブラリを選択することができる．本稿執筆時点で利用可能なライブラリは**表 4** に示す通りである．Intel コンパイラと親和性の高い Intel MPI，導入ベンダである SGI から提供される SGI MPT，InfiniBand ベンダである Mellanox から提供される Mellanox HPC-X に加えて，オープンソースの Open MPI，MVAPICH2 も選ぶことができる．

さらに Mellanox より，FCA (Fabric Collective Accelerator)[9]，さらに進んだ SHArP (Scalable Hierarchical Aggregation Protocol)[10] という Collective 通信の offload 機能が提供されている．FCA の機能は，HPC-X に加えて，SGI MPT でも利用可能である．また，Open MPI のコンパイル時にスイッチを指定して有効化できるが，今回利用した環境では有効にされていない．（HPC-X で提供されるのは，Mellanox によりカスタマイズされた Open MPI であり，Open MPI 1.10.3rc4 相当である．）加えて，低オーバヘッドのソフトウェアスタックの実装を目指した UCX (Unified Communication X)[11] も利用することができる．

3.4.2　PingPong 通信

IMB PingPong を用いて，2 ノード間の通信レイテンシ (**図 5**) とバンド幅 (**図 6**) を測定した．通信レイテンシの

表 4　Reedbush-U で利用可能な MPI ライブラリ

MPI ライブラリ	FCA 3.5	SHArP	UCX
Intel MPI 5.1.3	-	-	-
SGI MPT 2.14	○	-	-
Mellanox HPC-X 1.6.392	○	○	○
Open MPI 1.10.2	△	-	△
MVAPICH2 2.2rc1	(計画中?)		

最小値は UCX を有効にした HPC-X で，1.15μsec であった．しかし以前の測定で 0.88μsec を記録したこともあり，原因を調査中である．また，Intel MPI などにおいて，性能が安定しない現象が見られており，原因を調査する必要がある．

バンド幅は，MVAPICH2 を用いた場合に，最高で 4MB メッセージの場合に 11557MB/sec を得た．これは理論通信性能 100 Gbps の 92.5%である．MVAPICH2 ではメッセージ長全体に渡って安定した性能を示している．一方 SGI MPT では，他の 40%程度の性能しか得られていない．UCX を有効にした HPC-X においても，76%程度である．Intel MPI においては，64KB, 128KB で性能低下が見られる．

3.4.3　Allreduce 通信

続いて，IMB Allreduce により，420 ノードを用いた Allreduce 通信の実行時間 (**図 7**) を測定した．

SHArP を有効にした HPC-X では，64 byte まで 6μsec 前後で Allreduce が完了しており，SHArP が有効に機能していることがわかる．通常，多く用いられると考えられる，64 byte までの Allreduce を考えると，最も遅い Intel MPI に比べて，約 4 倍高速であった．また，FCA を有効にした HPC-X に比べても約 2 倍高速である．

一方，128 byte から 512 byte までは FCA を有効にした MPT が最も高速，1024 byte 以上では Intel MPI が最も高速，という結果が得られた．

SHArP 機能はつい最近リリースされたばかりであり，今後実アプリケーション上で FCA 機能とあわせて評価する必要がある．

3.5　MDTEST ベンチマーク

並列ファイルシステム上で，mdtest ベンチマークによりメタデータ性能を測定した．mdtest ベンチマークは，Larence Livermore National Lab. (LLNL) の Livermore Computing Center が公開している I/O ベンチマークの 1 つ [12] である．

今回は，ファイルの作成，stat, 削除について測定した．結果を**表 5** に示す．1 ノードの場合は 5000 回/秒以上，32 ノード 128 プロセスの場合は，同一ディレクトリ，個別ディレクトリいずれの場合もファイル作成には 10 万回/秒以上，削除には 6 万回/秒以上可能であった．

なお，Reedbush では，高速ファイルキャッシュシステムも，並列ファイルシステムと共通のメタデータによって管理されている．

表 5　mdtest の性能 (回/秒)

ノード数	ディレクトリ	File creation	File stat	File removal
1 ノード	同一	5149	5776	6129
32 ノード, 128 プロセス	同一	93481	154854	60338
	個別	111126	393736	68902

表 6　並列ファイルシステムにおける IOR 性能 (MB/sec)

I/O	R/W	420 ノード × 1 プロセス	420 ノード × 2 プロセス	420 ノード × 4 プロセス
MPIIO	Write	76519	76194	77962
	Read	77484	78568	79715
POSIX	Write	74308	77653	76774
	Read	79585	79944	80092

表 7　高速ファイルキャッシュシステムにおける IOR 性能 (MB/sec)

I/O	R/W	420 ノード × 1 プロセス	420 ノード × 2 プロセス
MPIIO	Write	217422	220005
	Read	196172	204259
POSIX	Write	218212	220852
	Read	196849	208592

3.6　IOR ベンチマーク

並列ファイルシステム，および高速ファイルキャッシュシステム上で IOR ベンチマークを実行し，I/O バンド幅を測定した．IOR ベンチマークは，mdtest と同様に LLNL の Livermore Computing Center が公開している I/O ベンチマークであり，ブロック入出力のスループットを計測するものである [13].

今回は，プロセス毎に異なるファイルに対する読み書きを測定した．並列ファイルシステムの結果を**表 6** に, 高速ファイルキャッシュシステムの結果を**表 7** に示す．1 回のシステムコールあたりの書き込みサイズは 1MiB とした．

いずれも，MPIIO を用いた場合と POSIX IO を用いた場合とで大きな性能差はない．並列ファイルシステムについては，最大 80 GB/sec, 高速ファイルキャッシュシステムについては，最大 220 GB/sec の性能が得られた．これらは，表 1 の理論ピーク性能と比較すると，それぞれ 55.2%, 50.5%となる．今回の結果は，十分飽和したところまで測定できていると言い切れないため，今後さらなる評価が必要である．

高速ファイルキャッシュシステムは新しい技術を用いたシステムであるため，どのような実アプリケーションが適しているのか，性能評価を行っていく必要がある．

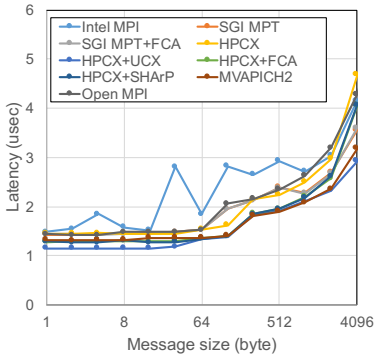

図 5　PingPong 通信のレイテンシ

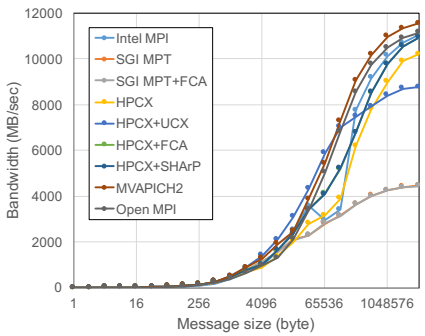

図 6　PingPong 通信のバンド幅

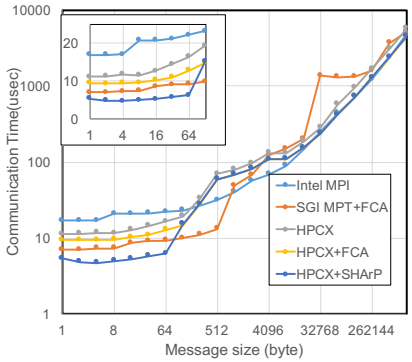

図 7　Allreduce 通信の実行時間

表 8　Poisson 3D で実施可能なケース

	番号付け	行列格納形式	外側ループ	その他
AR-0	Coaleased	CRS	行方向 (Row-wise)	
AR-1		Sliced-ELL		
AR-2				記憶容量削減型
AC-1			列方向 (Column-wise)	
AC-2				ブロック化
BR-0	Sequential	CRS	行方向 (Row-wise)	
BR-1		Sliced-ELL		
BC-1			列方向 (Column-wise)	
BC-2				ブロック化

4. 実アプリケーションコードに基づく性能評価

4.1 Poisson 3D

4.1.1 概要

Poisson 3D は有限体積法による三次元ポアソン方程式ソルバー [14], [15] から導かれる対称正定な疎行列を係数とする連立一次方程式を不完全コレスキー分解前処理付き共役勾配法 (Preconditioned Conjugate Gradient Method by Incomplete Cholesky Factorization，ICCG 法）によって解くプログラムであり，OpenMP によってマルチスレッドアーキテクチャ向けに並列化されている [14]．ICCG 法の不完全コレスキー分解，前進代入，後退代入のプロセスに現れるデータ依存性を回避するために RCM 法 (Reverse Cuthill-Mckee) に Cyclic マルチカラー法 (Cyclic Multicoloring, CM) を適用した CM-RC(m) 法（m: CM の色数）が使用されている [14]．

Poisson 3D では，(1) 番号付け (Coalesced, Sequential), (2) 行列格納形式 (CRS(Compressed Row Storage), Sliced-ELL [14])，(3) 外側ループ回転方向 (Row-wise, Column-Wise), (4) その他 （ブロック化，記憶容量削減型手法 [14] ）など様々な手法の組み合わせによるケースをテストすることが可能となっている（**表 8** 参照）．

4.1.2 計算機環境

本節では以下の 3 種類の計算機環境を使用した：

KNC: Intel Xeon Phi 5110P (Knights Corner)

表 9　各計算環境（1 ソケット）の概要

略称	KNC	IVB	BDW
名称	Intel Xeon Phi 5110P (Knights Corner)	Intel Xeon E5-2680 v2 (IvyBridge-EP)	Intel Xeon E5-2695 v4 (Broadwell-EP)
動作周波数 (GHz)	1.053	2.80	2.10
コア数（使用スレッド数）	60 (240)	10 (10)	18 (18)
理論演算性能 (GFLOPS)	1,010.9	224.0	604.8
主記憶容量 (GB)	8	32	128
理論メモリ性能 (GB/sec)	320	59.7	76.8
STREAM Triad (GB/sec)	159	49	64
キャッシュ構成	L1:32KB/core L2:512KB/core	L1:32KB/core L2:256KB/core L3:25MB/socket	L1:32KB/core L2:256KB/core L3:45MB/socket
コンパイルオプション	-O3 -qopenmp -mmic -align array64byte	-O3 -qopenmp -ipo -xAVX -align array32byte	-O3 -qopenmp -ipo -xCORE-AVX2 -align array32byte

IVB: Intel Xeon E5-2680v2 (IvyBridge-EP)

BDW: Intel Xeon E5-2695v4 (Broadwell-EP), Reedbush-U

プログラムは Fortran 90 で記述してあり，Intel Compiler (Ver. 16.0.3) / Intel Parallel Studio XE 2016 を使用した．**表 9** に計算機環境，コンパイルオプションの概要を示す．本節では各計算環境の 1 ソケットを使用した．

4.1.3 計算結果

表 8 に基づき，NX = NY = NZ = 128，総メッシュ数 2,097,152 の場合について検討を実施した．**図 8** に CM-RCM(m) の色数 m が 100 色までの BDW における計算結果（ICCG ソルバーの計算時間）を示す．**図 9** は各ハードウェアにおける CM-RCM(m) の最適色数（KNC：10 色，IVB，BDW：40 色）の場合の ICCG ソルバーの計算時間で

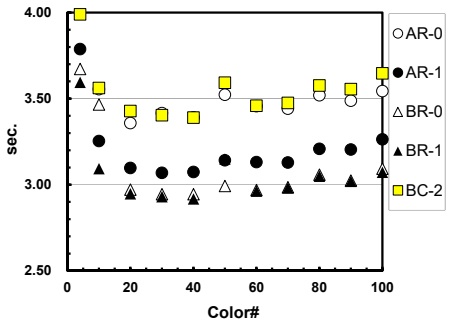

図 8 ICCG 法ソルバーの計算性能（CM-RCM(m) の色数と計算時間の関係），要素数 : 1283 (=2,097,152), Intel Xeon E5-2695 v4 (Broadwell-EP，BDW) による計算

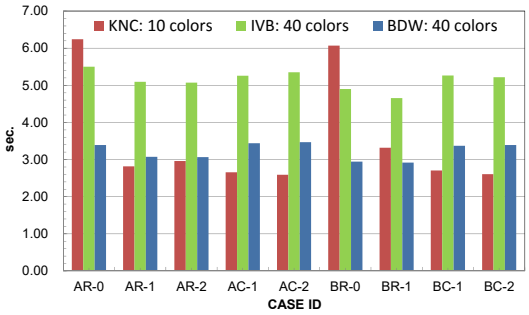

図 9 各ケースにおける ICCG 法ソルバーの計算性能（計算時間），要素数 : 1283 (=2,097,152), KNC: CM-RCM(10), IVB: CM-RCM(40), BDW: CM-RCM(40)

ある．また，**図 10** は各ハードウェアに対する最適なケース (KNC: BC-2, IVB, BDW: BR-1) における色数と計算時間の関係である．

全体的に Sequential reordering に Sliced-ELL を組み合わせた手法が良い性能を示している．外側ループ回転方向を列方向にし，更に CM-RCM の色毎にブロック化した BC-2 は，`!$omp simd` が適用されており，ベクトル化による性能向上を最も引き出しやすいため，KNC では最も高い性能を示す．一方，IVB，BDW では外側ループを行方向に回した BR-1 が最も良く，BC-2 ではベクトル化の効果が得られていないことがわかる．

KNC では CRS から Sliced-ELL に行列格納方法を変えた場合の性能向上 (AR-0⇒AR-1，BR-0⇒BR-1) が非常に顕著であるのに対して，IVB，BDW では非常に小さく，特に BDW では両者の差がほとんど無い．**図 11** は IVB，BDW で BR-0，BR-1 (Sequential + Row-wise，CRS (BR-0)，Sliced ELL (BR-1)) の計算性能（ICCG ソルバーの計算時間）を 4 色から 382 色（本問題設定における RCM 法の色数）まで比較したものである．BR-0 と BR-1 の差は BDW では IVB と比較して非常に小さいことがわかる．図 10 に示す KNC の場合においては，色数が増えると同期オーバーヘッドの増大のため，計算性能低下は顕著である．一方，IVB, BDW の場合は，色数の影響がほとんど無い．これは KNC では in-order 実行であるのに対し，IVB, BDW では out-of-order 実行であることによる影響と考えられる．

BDW は IVB と比較すると，ピーク性能で 2.70 倍，Stream の Triad 性能（メモリバンド幅）で 1.31 倍となっているが，ICCG 法の性能比は図 9〜11 によれば 1.60 倍程度である．Poisson 3D の ICCG ソルバーの 90%程度は疎行列計算（行列ベクトル積，前進後退代入）が占めるため，この比率は妥当と考えられる．BDW の最大性能が図 9, 10 における KNC の最大性能を下回っているのも同様の理由である．

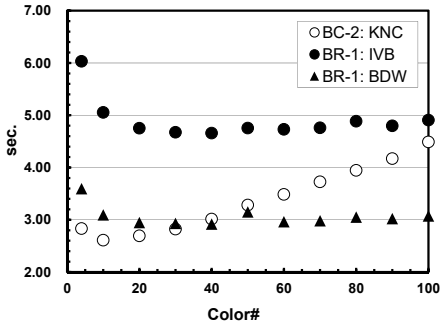

図 10 ICCG 法ソルバーの計算性能（色数と計算時間の関係），要素数 : 1283(=2,097,152), 各ハードウェアにおける最適ケース (KNC: BC-2, IVB, BDW: BR-1)

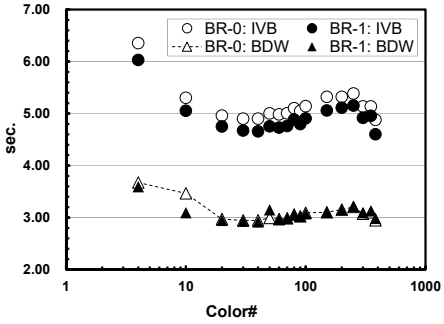

図 11 ICCG 法ソルバーの計算性能（色数と計算時間の関係），要素数 : 1283(=2,097,152), Intel Xeon E5-2680 v2 (IvyBridge-EP, IVB), Intel Xeon E5-2695 v4 (Broadwell-EP, BDW), BR-0 (CRS) · BR-1 (Sliced-ELL) の比較

4.2 GeoFEM/Cube 行列生成部

4.2.1 概要

並列有限要素法アプリケーションを元に整備した性能評価のためのベンチマークプログラム “GeoFEM/Cube” によって有限要素法における行列生成部の性能評価 [15], [16], [17], [18] を前節と同じプラットフォームの 1 ソケットを使用して実施した．本ベンチマークは，三次元弾性静解析問題（Cube 型モデル）に関する並列前処理付き反復法による疎行列ソルバーの実行時性能（GFLOPS 値）を様々な条件下で計測するものである．要素タイプは三次元一次六面体要素 (tri-linear) であり，各要素 8 つの節点を有している．プログラムは全て OpenMP ディレク

```
!$omp parallel (…)
    do color= 1, COLORtot
!$omp do
    do ip= 1, THREAD_num
      NBLK: calculated by (col_index, color, thread#)
    do ib= 1, NBLK
      do blk= 1, BLKSIZ
        icel: calculated by (col_index, ib, blk)
!$omp simd
        do je= 1, 8; do je= 1, 8
          <②要素行列成分の全体行列（疎行列）におけるアドレス探索＋格納>
        enddo; enddo
      enddo
      do blk= 1, BLKSIZ
        icel: calculated by (col_index, ib, blk)
        <①各積分点におけるヤコビアン，形状関数導関数計算>
!$omp simd
        do je= 1, 8; do je= 1, 8
          <③ガウス数値積分，要素行列成分計算＋格納>
        enddo; enddo
      enddo
      do blk= 1, BLKSIZ
        icel: calculated by (col_index, ib, blk)
!$omp simd
        do je= 1, 8; do je= 1, 8
          <④要素行列成分の全体行列への加算>
        enddo; enddo
      enddo
    enddo
    enddo
    enddo
!$omp end parallel
```

図 12 Type-A 実装の概要（六面体要素）[18]（COLORtot：要素色数 (=8), THREAD_NUM: スレッド数 (=240), 要素色数 (=8), col_index(color): 各色に含まれる要素数，NBLK: 要素ブロック総数，BLKSIZ: 要素ブロックサイズ，icel：要素番号），本ベンチマークでは`!$omp simd`は使用していない

ティヴを含む Fortran 90 および MPI で記述されている．GeoFEM で採用されている局所分散データ構造 [15] を使用しており，マルチカラー法等に基づくリオーダリング手法によりマルチコアプロセッサにおいて高い性能が発揮できるように最適化されている．また，MPI，OpenMP，Hybrid(OpenMP ＋ MPI) の全ての環境で稼動する．

三次元弾性静解析問題では係数行列が対称正定な疎行列となることから，前処理を施した共役勾配法 (CG) 法によって連立一次方程式を解いている．本来の GeoFEM/Cube ベンチマークでは前処理手法として Symmetric Gauss Seidel(SGS) を使用しているが，本研究では Block Diagonal Scaling 法 [18] を使用しており，OpenMP 並列化した場合の前処理プロセスにおけるデータ依存性を考慮する必要がないため，節点のリオーダリングは実施していない．また，三次元弾性問題では 1 節点あたり 3 つの自由度があるため，これらを 1 つのブロックとして取り扱っている．係数行列はこのブロック型の特性を利用したブロック CRS 形式によって格納されている．

本ベンチマークでは，先行研究 [18] で実施した計算例のうち，オリジナル実装（ケース O）の他，最も良い性能が得られた手法として，一定数（ブロックサイズ）の要素マトリクスをまとめて生成する Type-A 実装（ケース A，**図 12**）の 2 ケースを実施した．本ベンチマークでは図 12 の`!$omp simd`は挿入していない．

有限要素法では，要素毎に得られる積分方程式から導かれる密な要素行列を重ね合わせて疎な全体行列を生成する．係数行列生成のプロセスを OpenMP 等でスレッド並列化した場合，ある節点に複数の要素から同時にデータの書き込みが発生する場合がある．要素行列の重ね合わせを実施する際にはマルチカラーオーダリング等を使用してこ

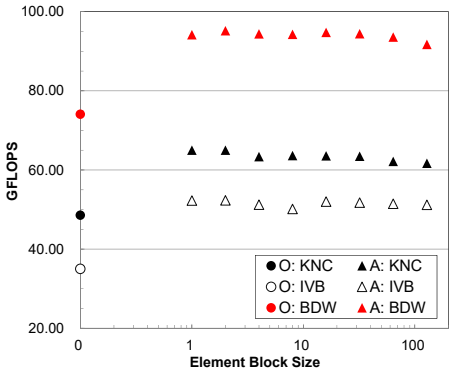

図 13 GeoFEM/Cube 計算結果，係数行列生成部計算性能

のような同時書き込みの発生を回避する方法が広く使用されており，図 12 に示した実装でもそのような手法が取り入れられている [18]．

4.2.2 計算例

本ベンチマークでは，Nx = Ny = Nz = 128 とした場合について検討した．したがって，節点数= 2,097,152 (=1283) であり，要素数は，六面体：2,048,383 (=1273), 四面体：12,290,298 (= 4 × 1273) である．またケース A におけるブロックサイズは 1〜128 とした．**図 13** に計算結果を示す．BDW による性能の基本的な傾向は KNC, IVB と変わらず，ケース A におけるブロックサイズは 64 以下では影響はほとんどなく，64 以上の場合やや性能が低下している．

BDW のケース O からケース A への性能向上率は約 28%であり，KNC（約 34%），IVB （約 49%）と比較して小さい．BDW の最適ケース（ケース A，ブロックサイズ =2）の計算性能は 95.2 GFLOPS（ピーク性能の 15.7%）であり， IVB の約 1.82 倍となっており，前節の Poisson 3D の場合よりも性能比は大きい．行列生成部は疎行列ソルバーと比較すると compute bound であるためと考えられる．また，KNC と比較しても約 1.46 倍と速度が向上している．現状では`!$omp simd` (図 12) を挿入したプログラムが IVB, BDW で正常に動作していないため，ここで示す性能は必ずしも各ハードウェアにおける最高性能ではない．KNC の場合は`!omp simd`の挿入によって約 1.45 倍の性能向上が得られている [18]．

4.3 GeoFEM/Cube 反復法ソルバー：各 MPI ライブラリの比較

本節では前節で述べた GeoFEM/Cube のオリジナルバージョン [16] に対して，Reedbush-U で利用可能な：

- Mellanox HPC-X
- Intel MPI
- MVAPICH
- OpenMPI

の 4 種類の MPI を適用し，Reedbush-U の全 420 ノードを使った性能測定を実施した．各ソケットに MPI プロセス

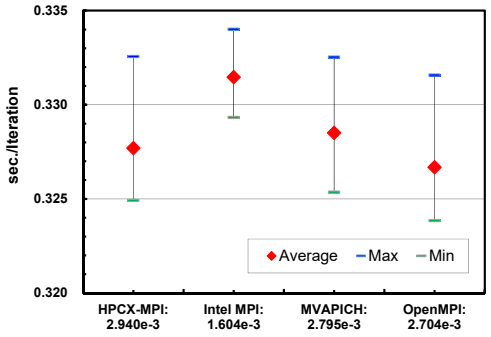

図 14 各 MPI による GeoFEM/Cube 性能比較（前処理付き CG 法ソルバー 1 反復当り計算時間），Reedbush-U 全系（420 ノード，840 ソケット）$2,800 \times 2,000 \times 1,200$ 節点（6.720×10^9 節点，2.016×10^{10} 自由度）六面体モデルを各方向 $14 \times 10 \times 6$ の計 840 領域に分割，各 MPI プロセスに割り当て，各 MPI プロセスのスレッド数は 18（＝コア数）

を割り当てているため，合計で 840 MPI プロセスとなる．前処理付き CG ソルバーの 1 反復あたりの計算時間によって比較している．

オリジナルの前処理は前節でも述べたように，SGS 法であり，局所化された前処理を Additive Schwartz Domain Decomposition によって安定化している [15], [16]．ノード内の並列化には RCM 法 (Reverse Cuthill-Mckee) を適用している．各反復において，集団通信 (MPI_Allreduce)，1 対 1 通信 (MPI_Isend / Irecv / Waitall) が使用されている．

本ベンチマークでは $2800 \times 2000 \times 1200$ 節点（6.720×10^9 節点，2.016×10^{10} 自由度）の六面体モデルを各方向 $14 \times 10 \times 6$ の計 840 領域に分割し，$200 \times 200 \times 200$ 節点（$= 2.400 \times 10^7$ 自由度）の部分領域を各 MPI プロセスに割り当てている．各 MPI プロセスのスレッド数は 18（＝コア数）であり，実行時には `numactl --localalloc` と同等のオプションをつけて実行している．**図 14** は各ケースにおいて 8 回測定を実施した際の前処理付き CG 法ソルバー計算時間（1 反復当り計算時間）の最大値，最小値，平均値，標準偏差である．ほぼ同等の性能であるが，Intel MPI がやや遅い．一方 Intel MPI は性能の振れ幅が他より小さく安定しているのが特徴である．今回は MPI プロセス当りの問題規模がやや大きく，通信のオーバーヘッドの影響がやや見えにくくなっているため，様々な問題規模でテストしてみる必要がある．また，8 回の計算はサンプル数としては少なめなため，より多くのケース数で実行する必要がある．

4.4 今後の課題

Poisson 3D BDW では，`!$omp simd` 挿入によるベクトル化の効果が充分得られておらず，検討が必要である．

行列生成 BDW，IVB では`!$omp simd` を挿入すると計算が正常に行われず，検討が必要である．

MPI MPI プロセスあたりの問題規模が小さいケースもやってみる必要がある．

5. おわりに

本稿では，2016 年 7 月に稼働を開始した Reedbush-U サブシステムを用いて性能評価を行った．Reedbush システムは，様々な最新技術を導入しており，コモディティ技術をベースにしていることから安定動作はしているものの，様々な性能最適化の余地がある．今後もシステムソフトウェアの最適化，プログラムの最適化についても進めていく必要がある．

Reedbush-U は，2 ヶ月間の試験運用期間を経て，2016 年 9 月から本運用を開始する予定である．また，Reedbush-H は 2017 年 3 月から稼働開始の予定で，その後 1 ヶ月の試験運用期間を経て，2017 年 4 月より全系による運用を開始する予定である．

Reedbush-H 稼働開始後には，本稿と同様にベンチマークにより GPU 搭載ノードについて性能評価を行う予定である．

謝辞 Reedbush システムの実験にご協力いただいた，日本 SGI 株式会社および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研究部門の皆様に感謝します．

参考文献

[1] 最先端共同 HPC 基盤施設：最先端共同 HPC 基盤施設（JCAHPC）について，最先端共同 HPC 基盤施設（オンライン），入手先 ⟨http://jcahpc.jp⟩（参照 2016-08-15）．

[2] : Whitepaper: NVIDIA Tesla P100, NVIDIA (online), available from ⟨https://images.nvidia.com/content/pdf/tesla/whitepaper/pascal-architecture-whitepaper.pdf⟩ (accessed 2016-08-15).

[3] McCalpin, J. D.: Memory Bandwidth and Machine Balance in Current High Performance Computers, *IEEE Computer Society Technical Committee on Computer Architecture (TCCA) Newsletter*, pp. 19–25 (1995).

[4] McCalpin, J. D.: STREAM Sustainable Memory Bandwidth in High Performance Computers, University of Virginia (online), available from ⟨http://www.cs.virginia.edu/stream/⟩ (accessed 2016-08-15).

[5] 大島聡史，實本英之，鴨志田良和，片桐孝洋，田浦健次朗，中島研吾：大規模超並列スーパーコンピューターシステム Oakleaf-FX(Fujitsu PRIMEHPC FX10) の性能評価，情報処理学会研究報告，Vol. 2012-HPC-135 (2012).

[6] Petitet, A., Whaley, R. C., Dongarra, J. and Cleary, A.: HPL - A Portable Implementation of the High-Performance Linpack Benchmark for Distributed-Memory Computers, ICL, University of Tennessee (online), available from ⟨http://www.netlib.org/benchmark/hpl/⟩ (accessed 2016-08-15).

[7] Dongarra, J., Heroux, M. and Luszczek, P.: HPCG Benchmark, ICL UT, SNL (online), available from ⟨http://www.hpcg-benchmark.org⟩ (accessed 2016-08-15).

[8] Gergana, S.: Getting Started with Intel MPI Benchmarks 4.1, Intel (online), available from ⟨https://software.intel.com/en-us/articles/intel-mpi-benchmarks⟩ (accessed 2016-08-16).

[9] : Fabric Collective Accelerator (FCA), Mellanox Technologies (online), available from ⟨http://www.mellanox.com/products/fca/⟩ (accessed 2016-08-16).
[10] : Mellanox Scalable Hierarchical Aggregation Protocol (SHArP), Mellanox Technologies (online), available from ⟨http://www.mellanox.com/page/products_dyn?product_family=261&mtag=sharp⟩ (accessed 2016-08-16).
[11] : Unified Communication X, Unified Communication X (online), available from ⟨http://www.openucx.org/Members/mellanox/⟩ (accessed 2016-08-16).
[12] : mdtest HPC Benchmark, Lawrence Livermore National Laboratory (online), available from ⟨https://sourceforge.net/projects/mdtest/⟩ (accessed 2016-08-16).
[13] : IOR HPC Benchmark, Lawrence Livermore National Laboratory (online), available from ⟨https://sourceforge.net/projects/ior-sio/⟩ (accessed 2016-08-16).
[14] 中島研吾：前処理付きマルチスレッド並列疎行列ソルバー，情報処理学会研究報告，Vol. HPC-139-6 (2013).
[15] GeoFEM: 並列有限要素法による固体地球シミュレーションプラットフォーム，RIST（オンライン），入手先 ⟨http://geofem.tokyo.rist.or.jp⟩（参照 2016-4-3）.
[16] Nakajima, K.: Parallel Iterative Solvers of GeoFEM with Selective Blocking Preconditioning for Nonlinear Contact Problems on the Earth Simulator, *ACM/IEEE Proceedings of SC2003* (2003).
[17] 中島研吾，片桐孝洋：マルチコアプロセッサにおけるリオーダリング付き非構造格子向け前処理付反復法の性能，情報処理学会研究報告，Vol. HPC-120-6 (2009).
[18] 中島研吾, 成瀬　彰, 大島聡史, 塙　敏博, 片桐孝洋, 田浦健次朗：有限要素法係数行列生成プロセスのメニィコア環境における最適化, 情報処理学会研究報告, Vol. HPC-152-12 (2015).